

FILEID**OUTPUTHLP

J 12

000000 UU UU TTTTTTTTTT PPPPPPPP UU UU TTTTTTTTTT HH HH LL PPPPPPPP
000000 UU UU TTTTTTTTTT PPPPPPPP UU UU TTTTTTTTTT HH HH LL PPPPPPPP
00 00 UU UU TT PP PP UU UU TT HH HH LL PP PP
00 00 UU UU TT PP PP UU UU TT HH HH LL PP PP
00 00 UU UU TT PP PP UU UU TT HH HH LL PP PP
00 00 UU UU TT PPPPPPPP UU UU TT HHHHHHHHHHH LL PPPPPPPP
00 00 UU UU TT PPPPPPPP UU UU TT HHHHHHHHHHH LL PPPPPPPP
00 00 UU UU TT PP PP UU UU TT HH HH LL PP
00 00 UU UU TT PP PP UU UU TT HH HH LL PP
00 00 UU UU TT PP PP UU UU TT HH HH LL PP
00 00 UU UU TT PP PP UU UU TT HH HH LL PP
000000 UUUUUUUUUU TT PP UUUUUUUUUU TT HH HH LLLLLLLL PP
000000 UUUUUUUUUU TT PP UUUUUUUUUU TT HH HH LLLLLLLL PP
.....
.....

LL IIIII SSSSSSS
LL IIIII SSSSSSS
LL IIIII SS
LL IIIII SS
LL IIIII SSSSSS
LL IIIII SSSSSS
LL IIIII SS
LL IIIII SS
LL IIIII SS
LL IIIII SSSSSS
LL IIIII SSSSSS

LBR
VO4

```
1      0001 0 MODULE lbr_outputhelp (
2      0002 0   LANGUAGE (BLISS32),
3      0003 0   IDENT = 'V04-000'
4      0004 0   )
5      0005 1 BEGIN
6      0006 1 %TITLE 'Prompting and library searching help function';
7      0007 1 ****
8      0008 1 *
9      0009 1 *
10     0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11     0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12     0012 1 * ALL RIGHTS RESERVED.
13     0013 1 *
14     0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15     0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16     0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17     0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18     0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19     0019 1 * TRANSFERRED.
20     0020 1 *
21     0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22     0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23     0023 1 * CORPORATION.
24     0024 1 *
25     0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26     0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27     0027 1 *
28     0028 1 *
29     0029 1 ****
30     0030 1 *
31     0031 1 ++
32     0032 1
33     0033 1 FACILITY: Library access procedures.
34     0034 1
35     0035 1 ABSTRACT:
36     0036 1
37     0037 1 LBR$OUTPUT_HELP outputs help text to a user specified output
38     0038 1 routine. This information is drawn from an explicitly named
39     0039 1 help library, or optionally, from user specified default help
40     0040 1 libraries. In addition, an optional prompting mode is available
41     0041 1 so that LBR$OUTPUT_HELP can interact with a user and continue to
42     0042 1 provide help information to a user after it has satisfied his
43     0043 1 initial help request.
44     0044 1
45     0045 1 ENVIRONMENT:
46     0046 1
47     0047 1 VAX native, user mode.
48     0048 1
49     0049 1 --
50     0050 1
51     0051 1
52     0052 1 AUTHOR: Peter George,          CREATION DATE: 01-May-1981
53     0053 1
54     0054 1 MODIFIED BY:
55     0055 1
56     0056 1 V03-005 GJA0081          Greg Awdziewicz        10-Apr-1984
57     0057 1 - Reduce the severity of the signaled "openin" error
```

: 58 0058 1 |
.: 59 0059 1 |
.: 60 0060 1 |
.: 61 0061 1 |
.: 62 0062 1 |
.: 63 0063 1 |
.: 64 0064 1 |
.: 65 0065 1 |
.: 66 0066 1 |
.: 67 0067 1 |
.: 68 0068 1 |
.: 69 0069 1 |
.: 70 0070 1 |
.: 71 0071 1 |
.: 72 0072 1 |
.: 73 0073 1 |
.: 74 0074 1 |
.: 75 0075 1 |
.: 76 0076 1 |
.: 77 0077 1 |
.: 78 0078 1 |
.: 79 0079 1 |
.: 80 0080 1 |
.: 81 0081 1 |
.: 82 0082 1 |
.: 83 0083 1 |
.: 84 0084 1 |
.: 85 0085 1 |
.: 86 0086 1 |
.: 87 0087 1 |
.: 88 0088 1 |
.: 89 0089 1 |
.: 90 0090 1 |--

from fatal to error in the open library routine.
- Return any error from open_library to the calling
program if the open fails for the specified main
library.
- Update the "ROUTINE VALUE:" comments to include the
codes "openin", "nohlplibs", and "usrinperr".

- V03-004 GJA0077 Greg Awdziewicz 19-Feb-1984
- Set prompt flags[hcf\$v_more] when switching libraries
at the end of the search_libs routine to fix AccVio
which was occurring when the subprompt for a topic which
is in more than one library comes from the subtopic in
other than the main library when this subtopic was
requested at the subtopic level from the main library.
- Flag the end-topic case to further gate the output
of the additional libraries info.
- Allow filenames to be 39 characters long.
- Change the routine Make_upper_case to use a translate
table instead of conditionals inside a loop.
- V03-003 JWT0066 Jim Teague 22-Nov-1982
Let RMS parse filenames.
- V03-002 PCG0014 Peter George 06-Oct-1982
Modify library name substring search so that it only
checks for substrings that start from the beginning of
the file name.
- V03-001 PCG0013 Peter George 01-Jul-1982
Make default search libraries work again by correcting
typographic error.

```

92      0091 1 LIBRARY 'SYSSLIBRARY:STARLET';
93      0092 1 REQUIRE 'PREFIX';
94      0231 1 REQUIRE 'LBRDEF';
95
96      0822 1 EXTERNAL ROUTINE
97          0824 1 lbr$close : ADDRESSING_MODE(GENERAL); ! Close library
98          0825 1 lbr$ini_control : ADDRESSING_MODE(GENERAL); ! Initialize librarian
99          0826 1 lbr$get_help : ADDRESSING_MODE(GENERAL); ! Get help text
100         0827 1 lbr$open : ADDRESSING_MODE(GENERAL); ! Open library
101
102         0829 1 FORWARD ROUTINE
103         0830 1 prompt_help, ! Main prompting loop
104         0831 1 search_libs, ! Search default libraries till help is found
105         0832 1 change_lib, ! Change current help lib context to specified library
106         0833 1 switch_libname, ! Update the library name descriptor
107         0834 1 tran_next_lib, ! Get next user specified default help library
108         0835 1 open_library, ! Open a help library
109         0836 1 close_library, ! Close a help library
110         0837 1 setup_keys, ! Set up help keys for lbr$get_help
111         0838 1 call_lbrhelp, ! Call lbr$get_help to get help from a library
112         0839 1 output_driver, ! Driver for the user supplied output routine
113         0840 1 libs_available, ! Output list of default libraries available
114         0841 1 file_present, ! Determine if file exists
115         0842 1 nohelp_log, ! Log unrecognized help requests
116         0843 1 remove_last_key, ! Remove the last help key in a list of keys
117         0844 1 remove_terminator, ! Remove the termination character at the end of a command
118         0845 1 make_upper_case; ! Make a string all upper case
119
120         0846 1
121         0847 1 EXTERNAL LITERAL
122             0848 1 lbr$endtopic,
123             0849 1 lbr$illoutrou,
124             0850 1 lbr$illinrou,
125             0851 1 lbr$nohlplibs,
126             0852 1 lbr$toomnyarg,
127             0853 1 lbr$usrinperr;
128
129         0854 1
130         0855 1 LITERAL
131             0856 1 filename_length = 39, ! Max filename length.
132             0857 1 main_libnumber = -2, ! Library number of /LIBRARY specified library
133             0858 1 external_libnumber = -1; ! Library number of non-default, non-/LIB library
134
135         0859 1 OWN
136             0860 1 end_topic_flag: BYTE
137             0861 1 sys$help : COUNTEDSTRING ('SYSSHLP:.HLB'),
138             0862 1 topic : COUNTEDSTRING ('Topic? '),
139             0863 1 subtopic : COUNTEDSTRING ('Subtopic? '),
140             0864 1 prompt_prefix : VECTOR [5, BYTE]
141             0865 1           INITIAL (BYTE (4, 13, 10, 13, 10)),
142             0866 1 control_flags : BBLOCK [hcf$sc_length]; ! Global control flags
143
144         0867 1
145             0868 1 BIND
146                 0869 1 help_flags = control_flags [hcf$ll_userlib], ! User default library flags
147                 0870 1 prompt_flags = control_flags [hcf$ll_prompt]: BLOCK[, BYTE]; ! Prompt control flags
148                 0871 1
149                 0872 1
150                 0873 1 MAP
151                     0874 1 help_flags : BBLOCK;
```

```
: 149      0875 1 %SBTTL 'Routine lbr$output_help';
: 150      0876 1 GLOBAL ROUTINE lbr$output_help(output_rout, output_size, keys_desc,
: 151          0877 1           library_desc, flags, input_rout) =
: 152      0878 2 BEGIN
: 153      0879 2
: 154      0880 2 ++
: 155      0881 2 FUNCTIONAL DESCRIPTION:
: 156      0882 2
: 157      0883 2 This routine is the entry point for this module. It performs
: 158      0884 2 some initial processing on the input parameters, opens the main
: 159      0885 2 help library, and then calls prompt_help to do all the real work.
: 160      0886 2
: 161      0887 2 CALLING SEQUENCE:
: 162      0888 2
: 163      0889 2 status = LBR$OUTPUT_HELP (output_routine, output_width [, [keys_desc]
: 164      0890 2           [, [library_desc] [, [flags] [, [input_routine]]]]])
: 165      0891 2
: 166      0892 2 INPUTS:
: 167      0893 2
: 168      0894 2     output_rout = address of user output routine
: 169      0895 2     output_size = width of the output line to be passed to the user
: 170      0896 2     output_routine
: 171      0897 2     keys_desc = address of string desc for keys
: 172      0898 2     library_desc = address of string desc for help library name
: 173      0899 2     flags = address of longword of option flags
: 174      0900 2     input_rout = address of user input routine
: 175      0901 2
: 176      0902 2 OUTPUTS:
: 177      0903 2
: 178      0904 2     The requested help text is passed to the output routine.
: 179      0905 2
: 180      0906 2 ROUTINE VALUE:
: 181      0907 2
: 182      0908 2     status lbr$_normal
: 183      0909 2     lbr$_illoutrou Output routine improperly specified or missing
: 184      0910 2     lbr$_illinrou Input routine improperly specified or missing
: 185      0911 2     lbr$_toomnyarg Too many arguments
: 186      0912 2     lbr$_nohlplibs No help libraries can be opened.
: 187      0913 2     lbr$_usrinperr Input error from user's action routine.
: 188      0914 2
: 189      0915 2
: 190      0916 2     Some errors which occur may be signaled. For example, if an error
: 191      0917 2     occurs while trying to open the specified main default library
: 192      0918 2     the error will be signaled as an "openin" error. In this case
: 193      0919 2     the return code value will have the inhibit bit set. So the
: 194      0920 2     return value will be
: 195      0921 2     shr$_openin OR hlp$C_facility OR sts$K_error OR sts$M_inhib_msg
: 196      0922 2
: 197      0923 2     |-- MAP
: 198      0924 2
: 199      0925 2     keys_desc : REF BBLOCK,
: 200      0926 2     library_desc : REF BBLÖCK;
: 201      0927 2
: 202      0928 2     BUILTIN
: 203      0929 2     ACTUALCOUNT,
: 204      0930 2     NULLPARAMETER;
: 205      0931 2
```

```

: 206      0932 2 LOCAL
: 207      0933 2   getcmd_line : BBLOCK [hlp$C_pagesize],
: 208      0934 2   getcmd_desc : BBLOCK [dsc$C_s_bln],
: 209      0935 2   indices : BBLOCK [hli$C_length],
: 210      0936 2   input_routine,
: 211      0937 2   librarystring : BBLOCK [nam$C_maxrss],
: 212      0938 2   libraryname : BBLOCK [dsc$C_s_bln],
: 213      0939 2   nomsg,
: 214      0940 2   output_routine,
: 215      0941 2   output_width,
: 216      0942 2   status;
: 217
: 218      0943 2 BIND
: 219      0944 2   main_libindex = indices [hlISL_mainindex],
: 220      0945 2   last_libindex = indices [hlISL_lastindex],
: 221      0946 2   last_libnumber = indices [hlISL_lastnum];
: 222
: 223      0947 2 help_flags = 0;
: 224
: 225      0948 2 IF ACTUALCOUNT() GTR hlp$C_params
: 226      0949 2 THEN RETURN lbr$_toomnyarg;
: 227
: 228      0950 2 IF NULLPARAMETER (hlp$C_outrout)
: 229      0951 2 OR ..output_rout EQL 0
: 230      0952 2 THEN RETURN lbr$_illoutrou
: 231      0953 2 ELSE output_routine = ..output_rout;
: 232
: 233      0954 2 IF NULLPARAMETER (hlp$C_outwidth)
: 234      0955 2 OR ..output_size EQL 0
: 235      0956 2 THEN output_width = hlp$C_liswidth
: 236      0957 2 ELSE output_width = ..output_size;
: 237
: 238      0958 2 getcmd_desc = 0;
: 239      0959 2 getcmd_desc [dsc$A_pointer] = getcmd_line;
: 240      0960 2 IF NOT NULLPARAMETER (hlp$C_linedesc)
: 241      0961 2 THEN BEGIN
: 242      0962 2   getcmd_desc [dsc$W_length] = .keys_desc [dsc$W_length];
: 243      0963 2   CHSMOVE (.getcmd_desc [dsc$W_length],
: 244      0964 2     .keys_desc [dsc$A_pointer],
: 245      0965 2     .getcmd_desc [dsc$A_pointer] );
: 246
: 247      0966 2 END;
: 248
: 249      0967 2 libraryname = 0;
: 250      0968 2 libraryname [dsc$A_pointer] = librarystring;
: 251      0969 2 IF NOT NULLPARAMETER (hlp$C_libname)
: 252      0970 2 THEN BEGIN
: 253      0971 2   help_flags [hlp$V_library] = true;
: 254      0972 2   libraryname [dsc$W_length] = .library_desc [dsc$W_length];
: 255      0973 2   CHSMOVE (.libraryname [dsc$W_length],
: 256      0974 2     .library_desc [dsc$A_pointer],
: 257      0975 2     .libraryname [dsc$A_pointer]);
: 258
: 259      0976 2 END
: 260      0977 2 ELSE help_flags [hlp$V_library] = false;
: 261
: 262      0978 2 IF NOT NULLPARAMETER (hlp$C_flags)
: 263      0979 2 THEN help_flags = ..help_flags
: 264      0980 2           OR (hlp$M_prompt AND ..flags)

```

```

: 263      0989 3          OR (hlp$M_process AND ..flags)
: 264      0990 3          OR (hlp$M_group AND ..flags)
: 265      0991 3          OR (hlp$M_system AND ..flags)
: 266      0992 3          OR (hlp$M_liblist AND ..flags)
: 267      0993 3          OR (hlp$M_help AND ..flags)
: 268      0994 2          ELSE help_flags = .help_flags OR hlp$M_prompt      ! else set defaults
: 269      0995 2          OR hlp$M_process OR hlp$M_group
: 270      0996 2          OR hlp$M_system AND NOT hlp$M_liblist
: 271      0997 2          AND NOT hlp$M_help;
: 272      0998 2
: 273      0999 2          IF .help_flags [hlp$V_prompt]           ! If prompting enabled
: 274      1000 3          THEN BEGIN
: 275      1001 4          IF ( NULLPARAMETER (hlp$C_inrout)        ! And output_routine missing
: 276      1002 4          OR .input_rout EQ 0 )
: 277      1003 3          THEN RETURN lbr$_illinrou
: 278      1004 3          ELSE input_routine = .input_rout;
: 279      1005 3          prompt_flags = hcf$M_cont;           ! Turn on prompting
: 280      1006 3          END
: 281      1007 2          ELSE prompt_flags = hcf$M_noprompt;       ! else turn off prompting
: 282      1008 2
: 283      1009 2          IF .help_flags [hlp$V_process] OR      ! If default lib searching enabled
: 284      1010 2          .help_flags [hlp$V_group] OR .help_flags [hlp$V_system]
: 285      1011 2          THEN help_flags = .help_flags OR hlp$M_all    ! then set all flag
: 286      1012 2          ELSE help_flags = .help_flags AND NOT hlp$M_all;  ! else clear all flag
: 287      1013 2
: 288      1014 2          IF .help_flags [hlp$V_library]           ! If library specified
: 289      1015 3          THEN BEGIN
: 290      1016 3          LOCAL
: 291      1017 3          local_status;
: 292      1018 3          nomsg = false;                      ! Signal error if library can't be opened
: 293      1019 3          local_status = open_library (main_libindex, libraryname, .nomsg); ! Open main library
: 294      1020 3          IF NOT .local_status THEN RETURN .local_status; ! Return any errors.
: 295      1021 3          last_libindex = .main_libindex;          ! Set last library used to main library
: 296      1022 3          last_libnumber = main_libnumber;
: 297      1023 3          END
: 298      1024 3
: 299      1025 2          ELSE BEGIN                                ! Else use a default library
: 300      1026 2
: 301      1027 2          LOCAL
: 302      1028 3          libno,
: 303      1029 3          acmode;
: 304      1030 2
: 305      1031 3          IF NOT .help_flags [hlp$V_all]           ! Are we allowed to?
: 306      1032 3          THEN RETURN lbr$_nohlplibs;
: 307      1033 3          libno = -1;                         ! If not then signal error
: 308      1034 3          status = false;                     ! Initialize search
: 309      1035 3          nomsg = true;                      ! Init while condition
: 310      1036 3          WHILE NOT .status                  ! Do not signal open errors
: 311      1037 4          DO BEGIN
: 312      1038 4          IF NOT tran next_lib (libraryname, acmode, libno) ! While more libraries
: 313      1039 4          THEN RETURN lbr$_nohlplibs;
: 314      1040 4          status = open_library (last_libindex, libraryname, .nomsg);
: 315      1041 3          END;
: 316      1042 3          last_libnumber = 0;                   ! Get and try to open one
: 317      1043 2
: 318      1044 2          END;                           ! Set lib number
: 319      1045 2          !

```

LBR_OUTPUTHELP Prompting and library searching help function D 13
V04=000 Routine [lbr\$output_help] 16-Sep-1984 02:04:00 VAX-11 Bliss-32 v4.0-742
[LBR.SRC]OUTPUTHELP.B32;1

Page 7
(3)

```
: 320      1046 2 ! Call prompt_help to do the real help work.  
: 321      1047 2 !  
: 322      1048 2 status = prompt_help(getcmd_desc, output_width, .input_routine, .output_routine,  
: 323          indices, libraryname);  
: 324      1050 2  
: 325      1051 2 IF .help_flags [hlp$v_library]           ! If library specified  
: 326          THEN close_library(main_libindex);       ! Close the main library  
: 327      1052 2  
: 328      1053 2 RETURN .status  
: 329      1054 2  
: 330      1055 2  
: 330      1056 1 END;                                ! Of lbr$output_help
```

```
:  
.TITLE LBR_OUTPUTHELP Prompting and library searching  
help function  
.IDENT \V04-000\  
.PSECT $OWNS,NOEXE,2
```

00000 END_TOPIC_FLAG:

```
00001    .BLKB 1  
00004    .BLKB 3  
00005    SYSHELP:.BYTE 13  
00012    .ASCII \SYSSHELP:.HLB\  
00014    TOPIC: .BYTE 2  
00015    .ASCII \Topic? \  
0001C    SUBTOPIC: .BYTE 7
```

```
0001D    .ASCII \Subtopic? \  
00027    .BLKB 1  
00028    PROMPT_PREFIX: .BYTE 10  
0002D    .BLKB 4, 13, 10, 13, 10  
00030    CONTROL_FLAGS: .BLKB 3  
00030    CONTROL_FLAGS: .BLKB 8
```

HELP_FLAGS= CONTROL_FLAGS
PROMPT_FLAGS= CONTROL_FLAGS+4
.EXTRN LBR\$CLOSE, [BR\$INI CONTROL
.EXTRN LBR\$GET HELP, LBR\$OPEN
.EXTRN LBR\$ENDTOPIC, LBR\$_ILLOUTROU
.EXTRN LBR\$_ILLINROU, LBR\$_NOHPLIBS
.EXTRN LBR\$_TOOMNYARG, LBR\$_USRINPERR

.PSECT \$CODE\$,NOWRT,2

57	0000' FCD8	00FC 00000	.ENTRY LBR\$OUTPUT_HELP, Save R2,R3,R4,R5,R6,R7 : 0876
5E		CF 9E 00002	MOVAB HELP_FLAGS, R7
		CE 9E 00007	MOVAB -808(SP), SP
		67 D4 0000C	CLRL HELP_FLAGS
06		6C 91 0000E	CMPB (AP), #6
		08 1B 00011	BLEQU 1\$
50	00000000G	8F D0 00013	MOVL #LBR\$_TOOMNYARG, R0 : 0952
		04 0001A	RET
		6C 95 0001B 1\$: 05 13 0001D	TSTB (AP) : 0954
			BEQL 2\$

LBR
V04

LBR_OUTPUTHELP Prompting and library searching help function 16-Sep-1984 02:04:00 VAX-11 Bliss-32 V4.0-742
V04-000 Routine lbr\$output_help 14-Sep-1984 12:37:45 [LBR.SRC]OUTPUTHELP.B32;1

Page 9
(3)

LBR
V04

08	50	00000000G	08	12	000FD	12\$:	BNEQ	13\$	#LBRS_ILLINROU, R0			1003
04	56	18	AC	D0	000FF	12\$:	MOVL	MOV	RET			1004
04	A7		01	D0	00106	13\$:	MOVL	INPUT ROUT.	INPUT_ROUTINE			1005
04	A7		04	D0	00107	13\$:	MOVL	#1. PROMPT_FLAGS				0999
04	A7		01	CE	00111	14\$:	BRB	15\$				1007
04	67		01	E0	00115	15\$:	MNEGL	#1. PROMPT FLAGS				1009
04	67		02	E0	00119		BBS	#1. HELP_FLAGS, 16\$			1010	
04	67		03	E1	0011D		BBS	#2. HELP_FLAGS, 16\$				
01	A7		08	88	00121	16\$:	BBC	#3. HELP_FLAGS, 17\$				
01	A7		04	11	00125		BISB2	#8. HELP_FLAGS+1			1011	
01	A7		08	8A	00127	17\$:	BRB	18\$			1012	
01	A7		02	E1	0012B	18\$:	BICB2	#8. HELP_FLAGS+1			1014	
21			52	D4	00130		BBC	#2. HELP_FLAGS+1, 19\$			1018	
			52	DD	00132		CLRL	NOMSG			1019	
		10	AE	9F	00134		PUSHL	NOMSG				
		011C	CE	9F	00137		PUSHAB	LIBRARYNAME				
0000V	CF		03	FB	0013B		PUSHAB	MAIN LIBINDEX				
	76		50	E9	00140		CALLS	#3. OPEN LIBRARY			1020	
0118	CE	0114	CE	D0	00143		BLBC	LOCAL STATUS, 26\$			1021	
011C	CE		02	CE	0014A		MOVL	MAIN LIBINDEX, LAST_LIBINDEX			1022	
			3D	11	0014F		MNEGL	#2. LAST_LIBNUMBER				
			03	E1	00151	19\$:	BRB	24\$			1014	
19	01	A7	01	CE	00156		BBC	#3. HELP_FLAGS+1, 21\$			1031	
	6E		52	01	7D	00159	MNEGL	#1. LIBNO			1033	
	2B		53	E8	0015C	20\$:	MOVQ	#1. NOMSG			1035	
		08	5E	DD	0015F		BLBS	STATUS, 23\$			1036	
		14	AE	9F	00161		PUSHL	SP			1038	
0000V	CF		03	AE	9F	00164	PUSHAB	ACMODE				
	08		50	FB	00167		PUSHAB	LIBRARYNAME				
	50	00000000G	8F	D0	0016F	21\$:	CALLS	#3. TRAN_NEXT_LIB			1039	
			04	00176			BLBS	R0, 22\$				
			52	DD	00177	22\$:	MOVL	#LBRS_NOHLPLIBS, R0				
			AE	9F	00179		RET					
0000V	CF		0120	CE	9F	0017C	PUSHL	NOMSG			1040	
	53		03	FB	00180		PUSHAB	LIBRARYNAME				
			50	D0	00185		PUSHAB	LAST_LIBINDEX				
			D2	11	00188		CALLS	#3. OPEN LIBRARY			1036	
		10	011C	CE	D4	0018A	BRB	RO, STATUS			1042	
			0C	AE	9F	0018E	23\$:	20\$			1048	
			0118	CE	9F	00191	CLRL	LAST_LIBNUMBER				
			0050	8F	BB	00195	PUSHAB	LIBRARYNAME				
			18	AE	9F	00199	PUSHAB	INDICES				
			0134	CE	9F	0019C	PUSHAB	#^M<R4,R6>				
09	0000V	CF	06	FB	001A0		PUSHAB	OUTPUT_WIDTH				
	53		50	D0	001A5		CALLS	GETCMD_DESC				
	01	A7	02	E1	001A8		MOVL	#6. PROMPT_HELP			1051	
	0000V	CF	0114	CE	9F	001AD	BBC	R0, STATUS			1052	
	50		01	FB	001B1		PUSHAB	#2. HELP_FLAGS+1, 25\$				
			53	D0	001B6	25\$:	CALLS	MAIN_LIBINDEX				
			04	001B9	26\$:		MOVL	#1. CLOSE LIBRARY			1054	
							RET	STATUS, R0			1056	

; Routine Size: 442 bytes. Routine Base: SCODES + 0000

```

: 332      1057 1 XSBITL 'Routine prompt_help';
: 333      1058 1 ROUTINE prompt_help (getcmd_desc, output_width, input_routine,
: 334                      output_routine, indices, libraryname) =
: 335      1059 1
: 336      1060 2 BEGIN
: 337      1061 2
: 338      1062 2 ++ FUNCTIONAL DESCRIPTION:
: 339      1063 2
: 340      1064 2 This routine contains the interactive code loop that is repeatedly
: 341      1065 2 executed when help is being run in prompting mode. This same loop
: 342      1066 2 is executed exactly once when noprompt is specified.
: 343      1067 2
: 344      1068 2
: 345      1069 2 INPUTS:
: 346      1070 2
: 347      1071 2     getcmd_desc = address of the descriptor for the set of keys
: 348      1072 2             to be processed
: 349      1073 2
: 350      1074 2     output_width = address of longword containing width of output line
: 351      1075 2
: 352      1076 2     input_routine = address of user supplied input routine
: 353      1077 2
: 354      1078 2     output_routine = addresss of user supplied output routine
: 355      1079 2
: 356      1080 2     indices = address of data structure containing indices of
: 357      1081 2             libraries currently in use
: 358      1082 2
: 359      1083 2     libraryname = address of string desc for default help library name
: 360      1084 2
: 361      1085 2 OUTPUTS:
: 362      1086 2
: 363      1087 2     None.
: 364      1088 2
: 365      1089 2 ROUTINE VALUE:
: 366      1090 2
: 367      1091 2     Always true.
: 368      1092 2
: 369      1093 2 -- MAP
: 370      1094 2     getcmd_desc : REF BBLOCK,
: 371      1095 2     indices : REF BBLOCK,
: 372      1096 2     libraryname : REF BBLOCK;
: 373      1097 2
: 374      1098 2 LOCAL
: 375      1099 2     char_pos,
: 376      1100 2     key_descs : VECTOR [dsc$c_s_bln * hlp$c_maxkeys, BYTE],
: 377      1101 2             !String descriptors for keys
: 378      1102 2     key_length_array : VECTOR [hlp$c_maxkeys],
: 379      1103 2             !Array of key lengths
: 380      1104 2     lib_name : BBLOCK [dsc$c_s_bln],
: 381      1105 2             !Library name descriptor
: 382      1106 2     lib_name_buf : VECTOR [filename_length, BYTE],
: 383      1107 2             !Library name buffer
: 384      1108 2     print_data : BBLOCK [hpd$c_length],
: 385      1109 2             !Data structure for output driver
: 386      1110 2     topic_prompt : BBLOCK [dsc$c_s_bln],
: 387      1111 2             !Topic prompt descriptor
: 388      1112 2     topic_prompt_buf : BBLOCK [hlp$c_pagesize],
: 389      1113 2             !Topic prompt buffer
: 390      1114 2     sub_prompt : BBLOCK [dsc$c_s_bln],
: 391      1115 2             !Sub-prompt descriptor
: 392      1116 2     sub_prompt_level,
: 393      1117 2             !Current key depth
: 394      1118 2     sub_prompt_line : BBLOCK [hlp$c_pagesize],
: 395      1119 2             !Sub-prompt line
: 396      1120 2     sub_prompt_buf : BBLOCK [hlp$c_pagesize],
: 397      1121 2             !Sub-prompt buffer
: 398      1122 2     status;
```

```

: 389 1114 2 BIND
: 390 1115 2 main_libindex = indices [hl$1_mainindex];
: 391 1116 2 last_libindex = indices [hl$1_lastindex];
: 392 1117 2 last_libnumber = indices [hl$1_lastnum];
: 393 1118 2 getcmd_line = .getcmd_desc [dsc$1a_pointer];
: 394 1119 2 true_keys = print_data [hpds$1_truekeys] : SIGNED BYTE;
: 395 1120 2 help_level = print_data [hpds$1_helplevel] : BYTE;
: 396 1121 2 print_flags = print_data [hpds$1_printflag] : BBLOCK;
: 397 1122 2 add_info_level = print_data [hpds$1_subpmtlev];
: 398 1123 2 sub_prompt_ptr = print_data [hpds$1_subpmtptr];
: 399 1124 2 length_array = print_data [hpds$1_lenarray] : REF VECTOR;
: 400 1125 2 outputroutine = print_data [hpds$1_outputrou];
: 401
: 402 1127 2 length_array = key_length_array;
: 403 1128 2 outputroutine = .output_routine;
: 404
: 405 1130 2 lib_name = 0;
: 406 1131 2 lib_name [dsc$1a_pointer] = lib_name_buf;
: 407 1132 2 IF NOT .help_flags [hlp$1_library]
: 408 1133 2 THEN switch_libname (.libraryname, lib_name);
: 409
: 410 1135 2 topic_prompt = 0;
: 411 1136 2 topic_prompt [dsc$1a_pointer] = topic_prompt_buf;
: 412 1137 2 CH$MOVE (.prompt_prefix [0], prompt_prefix [1], topic_prompt_buf);
: 413
: 414 1139 2 sub_prompt = 0;
: 415 1140 2 sub_prompt [dsc$1a_pointer] = sub_prompt_line;
: 416 1141 2 CH$MOVE (.prompt_prefix [0], prompt_prefix [1], sub_prompt_line);
: 417
: 418 1143 2 WHILE (.getcmd_desc [dsc$w_length] GTR 0)
: 419 1144 3 AND (CH$RCHAR (.getcmd_desc [dsc$1a_pointer]) EQL %' ')
: 420 1145 3 DO BEGIN
: 421 1146 3   getcmd_desc [dsc$w_length] = .getcmd_desc [dsc$w_length] - 1;
: 422 1147 3   getcmd_desc [dsc$1a_pointer] = .getcmd_desc [dsc$1a_pointer] + 1;
: 423 1148 2 END;
: 424
: 425 1150 3 IF (.getcmd_desc [dsc$w_length] GTR 0)
: 426 1151 3 AND (CH$RCHAR (.getcmd_desc [dsc$1a_pointer]) EQL %'a')
: 427 1152 3 THEN (change_lib (.getcmd_desc, .getcmd_desc [dsc$1a_pointer],
: 428 1153 3   .indices, lib_name));
: 429
: 430 1155 2 WHILE (.prompt_flags AND hcf$1m_cont) NEQ 0
: 431 1156 3 DO BEGIN
: 432
: 433 1158 3 IF .prompt_flags EQL hcf$1m_noprompt
: 434 1159 3 THEN prompt_flags = NOT hcf$1m_noprompt;
: 435
: 436 1161 3 IF .prompt_flags GTR hcf$1m_stay
: 437 1162 4 THEN BEGIN
: 438 1163 4 !
: 439 1164 4 ! If moving down a prompt level then update the subtopic prompt.
: 440 1165 4 !
: 441 1166 4 !
: 442 1167 4 !
: 443 1168 4 IF (.prompt_flags AND (hcf$1m_more OR hcf$1m_stay)) EQL hcf$1m_more
: 444 1169 5 THEN BEGIN
: 445 1170 5

```

: R


```

503      1228 5          THEN sub_prompt [dsc$w_length] = 0           ! then clear subtopic prompt
504      1229 6          ELSE BEGIN
505      1230 6              remove_last_key (sub_prompt,
506      1231 6                  key_length_array [.sub_prompt_level+1]);
507      1232 6                  prompt_flags = hcf$m_backup OR hcf$sm_cont; ! Set backup flag
508      1233 6              END
509      1234 6
510      1235 6          ELSE SELECTONE (CHSRCHAR (.getcmd_desc [dsc$w_length] +
511      1236 5              .getcmd_desc [dsc$sa_pointer] - 1)) OF SET      ! Test termination character
512      1237 5
513      1238 5          [%C'?']:
514      1239 6          BEGIN
515      1240 6              getcmd_desc [dsc$w_length] = .sub_length;
516      1241 6              getcmd_desc [dsc$sa_pointer] = getcmd_line;
517      1242 5              END;
518      1243 5
519      1244 5          [%X'1A']:
520      1245 5          EXITLOOP;
521      1246 5
522      1247 5          [OTHERWISE]:
523      1248 6          BEGIN
524      1249 6              getcmd_desc [dsc$w_length] = .getcmd_desc [dsc$w_length] + .sub_length;
525      1250 6              getcmd_desc [dsc$sa_pointer] = getcmd_line;
526      1251 5              END;
527      1252 5          TES:
528      1253 5
529      1254 5          remove_terminator (.getcmd_desc);           ! Remove termination character from
530      1255 5
531      1256 4          END;
532      1257 4
533      1258 4          ! If subtopic length equals zero, then do topic prompting
534      1259 4
535      1260 4          IF .sub_prompt [dsc$w_length] EQ 0
536      1261 4          THEN BEGIN
537      1262 5              getcmd_desc [dsc$w_length] = hlp$e_pagesize;       ! Init rest of descriptor
538      1263 5              getcmd_desc [dsc$sa_pointer] = getcmd_line;
539      1264 5
540      1265 5          IF .last_libnumber GEQ external_libnumber
541      1266 5              THEN BEGIN
542      1267 5                  CHSWCHAR (%C'a', topic_prompt_buf + 4);           ! If not main library
543      1268 5                  CHSMOVE (.lib_name [dsc$w_length],           ! Then insert library name in prompt
544      1269 6                      lib_name [dsc$sa_pointer],
545      1270 6                      topic_prompt_buf + 4 + 1);
546      1271 6                  CHSWCHAR (%X'20', topic_prompt_buf + 4 + 1 +
547      1272 6                      lib_name [dsc$w_length]);
548      1273 6                  CHSMOVE (.topic [0], topic [1],
549      1274 6                      topic_prompt_buf + 4 + 1 + .lib_name [dsc$w_length] + 1);
550      1275 6
551      1276 6                  topic_prompt [dsc$w_length] = 4 + 1 +
552      1277 6                      lib_name [dsc$w_length] + 1 + .topic [0];
553      1278 6
554      1279 6
555      1280 6          END
556      1281 6          ELSE BEGIN
557      1282 6              topic_prompt [dsc$w_length] = 4 + .topic [0];   ! Otherwise, do not include library
558      1283 6              CHSMOVE (.topic [0], topic [1],
559      1284 6                  topic_prompt_buf + 4);

```

```

560      1285 5           END;
561
562      1286 5
563      1287 6           IF (.status = (.input_routine) (.getcmd_desc [dsc$w_length],
564                                topic_prompt, .getcmd_desc [dsc$w_length])) EQL RMSS_EOF
565      1288 5           THEN EXITLOOP
566      1289 5           ELSE IF NOT .status
567      1290 5           THEN RETURN lbr$_usrinperr;
568      1291 5
569      1292 5
570      1293 7           IF ((char_pos = CHSFIND NOT CH (.getcmd_desc [dsc$w_length],
571                                .getcmd_desc [dsc$w_pointer], %ASCII ',')) EQL 0)
572      1294 6           OR (CHSRCHAR (.char_pos) EQL %X'0D')
573      1295 6           THEN (IF (.last_libnumber LSS external.libnumber
574                                OR NOT .help_flags [hlp$v_library])
575                                THEN EXITLOOP
576                                ELSE BEGIN
577                                close_library (last_libindex);
578                                last_libnumber = main.libnumber;
579                                last_libindex = main.libindex;
580                                lib_name [dsc$w_length] = 0;
581                                getcmd_desc [dsc$w_length] = 0;
582                                getcmd_desc [dsc$w_pointer] = getcmd_line;
583                                prompt_flags = hcf$w_backup OR hcf$w_cont;
584                                END;)

585      1308 6           ELSE SELECTONE (CHSRCHAR (.getcmd_desc [dsc$w_length] +
586                                .getcmd_desc [dsc$w_pointer] - 1)) OF SET
587      1309 6
588      1310 5           ! Test termination character
589      1311 5
590      1312 5           ! If ?
591      1313 6           ! Then repeat help for old library
592
593      1314 6
594      1315 6
595      1316 5
596      1317 5
597      1318 5           ! If CNTL-Z
598      1319 5           ! Then get out of help
599      1320 5
600      1321 5           EXITLOOP;
601
602      1322 6           ! If help keys
603      1323 6           ! And start with 'a'
604      1324 5           ! Then change to specified library
605      1325 5
606      1326 5
607      1327 5           ! Remove terminator from
608      1328 5
609      1329 4
610      1330 4
611      1331 3           END;
612
613      1332 3           ! Of prompt for keywords
614
615      1333 3           IF (.prompt_flags AND hcf$w_backup) EQL 0
616      1334 4           THEN BEGIN
617      1335 5           IF .prompt_flags NEQ (NOT hcf$w_noprompt)
618                                THEN prompt_flags = hcf$w_cont;
619      1336 4           sub_prompt_ptr = sub_prompt_buf;
620      1337 4           help_level = 0;
621      1338 4           add_info_level = 0;
622      1339 4           setup_keys (.getcmd_desc, key_descs, true_keys);
623      1340 4           print_flags [hpd$w_init] = 0;
624
625      1341 4

```

```

617   1342 4    end_topic_flag = false;                                ! Clear flag which aborts text.
618   1343 5    IF NOT (.status = search_libr (key_descs, .output_width, print_data, .indices, lib_name, .sub_prompt [dsc$w_length])) ! Search the libraries for help
619   1344 5
620   1345 4    THEN RETURN .status;
621   1346 4    IF .print_flags [hpds$ init]
622   1347 4    THEN nohelp_log (.getcmd_desc);
623   1348 4    IF .help_flags [hlp$ liblist] AND
624   1349 4    NOT .end_topic_flag AND
625   1350 5    ((.true_keys LEQ 0) OR
626   1351 5    ((.add_info_level EQL 0) AND .print_flags [hpds$ init]))
627   1352 4    THEN
628   1353 4    libs_available (.output_routine, .output_width);
629   1354 4
630   1355 3 ELSE prompt_flags = hcf$m_cont OR hcf$m_stay;           ! Say to continue prompting at current
631   1356 3
632   1357 2 END;                                                 ! Of while
633   1358 2
634   1359 2 RETURN true
635   1360 2
636   1361 1 END;                                               ! Of prompt_help

```

OFFC 00000 PROMPT_HELP:						
	SE	F92C	CE 9E 00002	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 1058
	5A	14	AC D0 00007	MOVL	-1748(SP), SP	: 1115
	58	04	AC D0 0000B	MOVL	INDICES, R10	: 1118
	59	04	A8 9E 0000F	MOVAB	GETCMD_DESC, R8	
			69 DD 00013	PUSHL	4(R8), R9	
					(R9)	
	FF50	CD	88 AD 9E 00015	MOVAB	KEY LENGTH ARRAY, LENGTH ARRAY	: 1127
	FF4C	CD	10 AC D0 0001B	MOVL	OUTPUT ROUTINE, OUTPUTROUTINE	: 1128
			80 AD D4 00021	CLRL	LIB NAME	: 1130
OB	84	AD	FF58 CD 9E 00024	MOVAB	LIB_NAME_BUF, LIB_NAME+4	: 1131
	0000'	CF	02 E0 0002A	BBS	#2, HELP_FLAGS+1, -1\$: 1132
			80 AD 9F 00030	PUSHAB	LIB NAME	: 1133
			18 AC DD 00033	PUSHL	LIBRARYNAME	
	0000V	CF	02 FB 00036	CALLS	#2, SWITCH_LIBNAME	
			FF3C CD D4 0003B	CLRL	TOPIC PROMPT	
	FF40	CD	FD3C CD 9E 0003F	MOVAB	TOPIC_PROMPT BUF, TOPIC_PROMPT+4	: 1135
		50	0000'	MOVZBL	PROMPT PREFIX, R0	: 1136
FD3C	CD	0000'	CF 9A 00046	MOVC3	R0, PROMPT_PREFIX+1, TOPIC_PROMPT_BUF	: 1137
			FD34 CD D4 00053	CLRL	SUB_PROMPT	: 1139
	FD38	CD	020C CE 9E 00057	MOVAB	SUB_PROMPT LINE, SUB_PROMPT+4	: 1140
		50	0000'	MOVZBL	PROMPT PREFIX, R0	: 1141
020C	CE	0000'	CF 9A 0005E	MOVC3	R0, PROMPT_PREFIX+1, SUB_PROMPT_LINE	
			50 28 00063	TSTW	(R8)	
			68 B5 0006B	BEQL	3\$	
		20	00 B9 91 0006F	CMPB	20(R9), #32	: 1143
			06 12 00073	BNEQ	3\$	
			68 B7 00075	DECW	(R8)	: 1144
			69 D6 00077	INCL	(R9)	: 1146
			F0 11 00079	BRB	2\$: 1147
			68 B5 0007B	TSTW	(R8)	: 1143
			15 13 0007D	BEQL	4\$: 1150

LBR
V04

LBR_OUTPUTHELP Prompting and library searching help function N 15
V04-000 Routine prompt_help 16-Sep-1984 02:04:00 VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1 14-Sep-1984 12:37:45

Page 17
(4)

LBR
V04

00	B9	04	AE	50	D0	00167	MOVL	R0, STATUS						
		04	8F	51	D1	00168	CMPBL	STÁTUS, #98938						
		03		51	13	00173	BEQL	20\$						
		04		AE	E8	00175	BLBS	STATUS, 14\$						
				00D1	31	00179	BRW	28\$						
				20	3B	0017C	14\$: SKPC	#32, (R8), 20(R9)						
				02	12	00181	BNEQ	15\$						
				51	D4	00183	CLRL	R1						
		08	AE	51	D5	00185	MOVL	R1, CHAR_POS						
				06	13	00189	BEQL	16\$						
		0D		BE	91	0018B	CMPB	@CHAR_POS, #13						
				1E	12	0018F	BNEQ	18\$						
				5B	D7	00191	DECL	SUB_PROMPT_LEVEL						
				06	12	00193	BNEQ	17\$						
			FD34	CD	B4	00195	CLRW	SUB_PROMPT						
				33	11	00199	BRB	22\$						
				8C	AD4B	DD	0019B	17\$: PUSHL	KEY_LENGTH_ARRAY+4[SUB_PROMPT_LEVEL]					
			FD34	CD	9F	0019F	PUSHAB	SUB_PROMPT						
		0000V	CF	02	FB	001A3	CALLS	#2, REMOVE_LAST_KEY						
		0000'	CF	11	D0	001A8	MOVL	#17, PROMPT_FLAGS						
				1F	11	001AD	BRB	22\$						
		50		68	3C	001AF	MOVZWL	(R8), R0						
		50		69	CO	001B2	ADDL2	(R9), R0						
		50		A0	9A	001B5	MOVZBL	-1(R0), R0						
		3F		50	91	001B9	CMPB	R0, #63						
				05	12	001BC	BNEQ	19\$						
		68		56	B0	001BE	MOVW	SUB_LENGTH, (R8)						
				08	11	001C1	BRB	21\$						
		1A		50	91	001C3	19\$: CMPB	R0, #26						
				7F	13	001C6	BEQL	27\$						
		68		56	A0	001C8	ADDW2	SUB_LENGTH, (R8)						
		69		6E	D0	001CB	MOVL	(SP), (R9)						
				58	DD	001CE	21\$: PUSHL	R8						
		0000V	CF	01	FB	001D0	CALLS	#1, REMOVE_TERMINATOR						
			FD34	CD	B5	001D5	TSTW	SUB_PROMPT						
				03	13	001D9	BEQL	24\$						
				00F6	31	001DB	BRW	37\$						
		68		0200	8F	BO	001DE	24\$: MOVW	#512, (R8)					
					6E	DO	001E3	MOVL	(SP), (R9)					
		FFFFFFFFFF	8F	0000'	CF	9A	001E6	MOVZBL	TOPIC, R7					
				08	AA	D1	001EB	CMPB	8(R10), #1					
					2C	19	001F3	BLSS	25\$					
			FD40	CD	40	8F	90	001F5	MOVBL	#64, TOPIC_PROMPT_BUF+4				
				56	80	AD	3C	001FB	MOVZWL	LIB_NAME, R6				
				84	BD	56	28	001FF	MOVCL3	R6, ALIB NAME+4, TOPIC_PROMPT_BUF+5				
			FD41	CD46			20	90	00206	MOVBL	#32, TOPIC_PROMPT_BUF+5[R6]			
			0000'	CF		57	28	0020C	MOVCL3	R7, TOPIC+T, TOPIC_PROMPT_BUF+6[R6]				
				50		06	A746	9E	00215	MOVAB	6(R7)[R6], R0			
			FF3C	CD			50	BO	0021A	MOVW	R0, TOPIC_PROMPT			
						0E	11	0021F	BRB	26\$				
						04	A1	00221	25\$: ADDW3	#4, R7, TOPIC_PROMPT				
						57	28	00227	MOVC3	R7, TOPIC+1, TOPIC_PROMPT_BUF+6				
						58	DD	0022F	26\$: PUSHL	R8				
						CD	9F	00231	PUSHAB	TOPIC_PROMPT				
						58	DD	00235	PUSHL	R8				
						03	FB	00237	CALLS	#3, @INPUT_ROUTINE				
						50	DO	0023B	MOVL	R0, STATUS				

LBR_OUTPUTHELP Prompting and library searching help function 8 14
 V04=000 Routine prompt_help 16-Sep-1984 02:04:00 14-Sep-1984 12:37:45 VAX-11 Bliss-32 V4.0-742
 [LBR.SRC]OUTPUTHLP.B32;1 Page 18 (4)

		0001827A	8F	04	AE	D1	0023F	27\$:	CMPL	STATUS, #98938		
			08	04	AE	E8	00249	28\$:	BEQL	32\$		
			50	00000000G	BF	D0	0024D	28\$:	BLBS	STATUS, 29\$	1290	
						04	00254	RET	MOVL	#LBR\$_USRINPERR, R0	1291	
00	B9		68		20	38	00255	29\$:	SKPC	#32, (R8), 30(R9)	1293	
					02	12	0025A		BNEQ	30\$		
		08	AE		51	D4	0025C		CLRL	R1		
					06	13	00262	30\$:	MOVL	R1, CHAR_POS	1294	
		0D		08	BE	91	00264		BEQL	31\$, @CHAR_POS, #13	1295	
		FFFFFFF	BF	08	32	12	00268		CMPB			
					AA	D1	0026A	31\$:	BNEQ	34\$, B(R10), #-1	1296	
					03	18	00272		CMPL			
F7	0000'	CF		08	D0F9	31	00274	32\$:	BGEQ	33\$, BRW		
					02	E1	00277	33\$:	BBC	45\$, #2, HELP_FLAGS+1, 32\$	1297	
	0000V	CF		04	AA	9F	0027D		PUSHAB	4(R10)	1300	
	08	AA			01	F8	00280		CALLS	#1, CLOSE_LIBRARY		
	04	AA			02	CE	00285		MNEG	#2, 8(R10)	1301	
			80		6A	D0	00289		MOVL	(R10), 4(R10)	1302	
					AD	B4	0028D		CLRW	LIB_NAME	1303	
					68	B4	00290		CLRW	(R8)	1304	
	0000'	69			6E	D0	00292		MOVL	(SP), (R9)	1305	
		CF			11	D0	00295		MOVL	#17, PROMPT_FLAGS	1306	
			50		31	11	0029A	34\$:	BRB	36\$,	1293	
			50		68	3C	0029C		MOVZWL	(R8), R0	1310	
			50		69	C0	0029F		ADDL2	(R9), R0		
			3F	FF	A0	9A	002A2		MOVZBL	-1(R0), R0	1312	
					50	91	002A6		CMPB	R0, #63		
					07	12	002A9		BNEQ	35\$,		
					68	B4	002AB		CLRW	(R8)	1314	
		69			6E	D0	002AD		MOVL	(SP), (R9)	1315	
					18	11	002B0		BRB	36\$,	1309	
		1A			50	91	002B2	35\$:	CMPB	R0, #26	1318	
			40	8F	08	BE	91	002B7	BEQL	32\$,		
					0F	12	002BC		CMPB	@CHAR_POS, #64	1322	
					80	AD	9F	002BE	BNEQ	36\$,		
					5A	DD	002C1		PUSHAB	LIB_NAME	1323	
					10	AE	DD	002C3	PUSHL	R10	1324	
					58	DD	002C6		PUSHL	CHAR_POS	1323	
	0000V	CF			04	FB	002C8		PUSHL	R8		
					58	DD	002CD	36\$:	CALLS	#4, CHANGE_LIB		
	03	0000V	CF		01	FB	002CF		PUSHL	R8	1327	
		0000'	CF		04	E1	002D4	37\$:	CALLS	#1, REMOVE_TERMINATOR		
					0088	31	002DA		BBC	#4, PROMPT_FLAGS, 38\$	1333	
					0000'	CF	D5	002DD	38\$:	BRW	43\$,	
					05	13	002E1		TSTL	PROMPT_FLAGS	1335	
		0000'	CF		01	D0	002E3		BEQL	39\$,		
	FF44	CD			AE	9E	002E8	39\$:	MOVL	#1, PROMPT_FLAGS	1336	
			OC		FF55	CD	94	002EE	MOVAB	SUB_PROMPT_BUF, SUB_PROMPT_PTR	1337	
			FF48		FF54	CD	D4	002F2	CLRB	HELP_LEVEL	1338	
					CD	9F	002F6		CLRL	ADD_INFO_LEVEL	1339	
			80		AD	9F	002FA		PUSHAB	TRUE_KEYS	1340	
					58	DD	002FD		PUSHAB	KEY_DESCS		
	0000V	CF			03	FB	002FF		PUSHL	R8		
		FF56	CD		01	8A	00304		CALLS	#3, SETUP_KEYS		
									BICB2	#1, PRINT_FLAGS	1341	

; Routine Size: 884 bytes, Routine Base: \$CODES + 01BA

```

: 638    1362 1 XSBTTL 'Routine search_libs';
: 639    1363 1 ROUTINE search_libs (keydescs, outputwidth, printdata,
: 640          indices, libname, subpromptlen) =
: 641    2 BEGIN
: 642    2
: 643    2 ++
: 644    2 : FUNCTIONAL DESCRIPTION:
: 645    2
: 646    22 : This routine searches the default help libraries and calls the
: 647          librarian function to extract help from each help library as
: 648          required.
: 649    2
: 650    22 : INPUTS:
: 651    22
: 652    1376 22 keydescs = address of vector of key descriptors
: 653    1377 22
: 654    1378 22 outputwidth = address of longword containing width of output line
: 655    1379 22
: 656    1380 22 printdata = address of data structure containing info for
: 657          the output driver
: 658    1381 22
: 659    1382 22
: 660    1383 22 indices = address of data structure containing indices of
: 661          libraries currently in use
: 662    1384 22
: 663    1385 22
: 664    1386 22 libname = address of descriptor for the user default library name
: 665    1387 22
: 666    1388 22 subpromptlen = total length of help keys, zero => at topic prompt level
: 667    1389 22
: 668    1390 22 : OUTPUTS:
: 669    1391 22
: 670    1392 22 printdata = flags are manipulated by this routine and other
: 671          values are altered by the output driver
: 672    1393 22
: 673    1394 22
: 674    1395 22 indices = updated to reflect library that help information
: 675          was eventually extracted from
: 676    1396 22
: 677    1397 22
: 678    1398 22 libname = if help found in default user library, updated to
: 679          the file name of that library
: 680    1399 22
: 681    1400 22
: 682    1401 22 : ROUTINE VALUE:
: 683    1402 22
: 684    1403 22 Always true.
: 685    1404 22
: 686    1405 22 :-- MAP
: 687    1406 22 indices : REF BBLOCK,
: 688    1407 22 keydescs : REF BBLOCK,
: 689    1408 22 libname : REF BBLOCK,
: 690    1409 22 printdata : REF BBLOCK;
: 691    1410 22
: 692    1411 22 : BIND
: 693    1412 22 main_libindex = indices [hl$1_mainindex], ! Index of /LIB library
: 694    1413 22 last_libindex = indices [hl$1_lastindex], ! Index of last library examined
: 695    1414 22 last_libnumber = indices [hl$1_lastnumb], ! No. of last library examined, relative to all defa
: 696    1415 22 print_flags = printdata [hpdsb_printflag] : BBLOCK; ! Flags for output driver
: 697    1416 22
: 698    1417 22
: 699    1418 22 LOCAL

```

```

695    1419 2      current_libindex,  

696    1420 2      current_libnumber,  

697    1421 2      deflib_acmode : BYTE,  

698    1422 2      librarystring : BBLOCK [namSc_maxrss],  

699    1423 2      libraryname : BBLOCK [dsc$C_s_bln],  

700    1424 2      nomsg,  

701    1425 2      user_ibno,  

702    1426 2      status;  

703  

704    1428 2      nomsg = true;  

705    1429 2      print_flags [hpdsV_found] = false;  

706    1430 2      libraryname [dsc$A_pointer] = librarystring;  

707  

708    1432 2      IF (.subpromptlen)  

709    1433 2          AND (.help_flags AND hlp$M_all) NEQ 0)  

710    1434 2          AND (.keydescs [dsc$W_length] NEQ 0)  

711    1435 2          THEN print_flags [hpdsV_all] = false  

712    1436 2          ELSE print_flags [hpdsV_all] = true;  

713  

714    1438 3      IF NOT (status = call_lbrhelp (last_libindex,  

715    1439 3          .outputwidth, .printdata, .keydescs))  

716    1440 2      THEN RETURN (.status);  

717  

718    1442 2      IF .print_flags [hpdsV_found]  

719    1443 3          THEN BEGIN  

720    1444 3              print_flags [hpdsV_found] = false;  

721    1445 3              RETURN true;  

722    1446 2          END;  

723  

724    1448 2      print_flags [hpdsV_all] = false;  

725  

726    1450 2      IF .help_flags [hlp$V_library]  

727    1451 3          AND (.last_libnumber GEQ 0)  

728    1452 3          THEN BEGIN  

729    1453 3              current_libindex = .main_libindex;  

730    1454 4              IF NOT (status = call_lbrhelp (last_libindex,  

731    1455 4                  .outputwidth, .printdata, .keydescs))  

732    1456 3              THEN RETURN (.status);  

733    1457 2          END;  

734  

735    1459 2      IF .print_flags [hpdsV_found]  

736    1460 2          THEN current_libnumber = main_libnumber  

737    1461 2          ELSE current_libnumber = -1;  

738  

739    1463 2      user_ibno = -1;  

740    1464 2      WHILE (NOT .print_flags [hpdsV_found]) AND  

741    1465 3          (tran_nextLib (libraryname, deflib_acmode, user_ibno) NEQ 0)  

742    1466 3      DO BEGIN  

743    1467 3          current_libnumber = .current_libnumber + 1;           ! Increment current libnumber  

744    1468 3          IF .current_libnumber NEQ .last_libnumber           ! If lib not already open and searched  

745    1469 4          THEN IF Topen_library (current_libindex, libraryname, .nomsg)   ! If library successfully opened  

746    1470 4          THEN BEGIN  

747    1471 4              status = call_lbrhelp (current_libindex,  

748    1472 4                  .outputwidth, .printdata, .keydescs);  

749    1473 4              IF NOT .print_flags [hpdsV_found]                ! If help not found  

750    1474 4                  THEN close_library (current_libindex);        ! Then close library  

751    1475 4              IF NOT .status THEN RETURN (.status);           ! If error then return

```

LBR
V04

```

752 1476 3 END:
753 1477 2
754 1478 1
755 1479 2 IF NOT .print_flags [hpdsV_found]
756 1480 3 THEN BEGIN
757 1481 3   current_libindex = .last_libindex;
758 1482 3   current_libnumber = .last_libnumber;
759 1483 3   print_flags [hpdsV_all] = true;
760 1484 4   IF NOT (status = call_lbrhelp (last_libindex,
761 1485 4     outputwidth, .printdata, .keydescs))
762 1486 3     THEN RETURN (.status);
763 1487 2
764 1488 2
765 1489 3 IF (.current_libnumber NEQ .last_libnumber)
766 1490 3 THEN BEGIN
767 1491 4   IF (.last_libnumber GTR external_libnumber)
768 1492 3     THEN close_library (last_libindex);
769 1493 4   IF (.current_libnumber GTR external_libnumber)
770 1494 3     THEN
771 1495 4       BEGIN
772 1496 4         switch_libname (libraryname, .libname);
773 1497 4         prompt_flags[hcf$V_more] = true;
774 1498 4       END
775 1499 3     ELSE
776 1500 3       libname [dsc$w_length] = 0;
777 1501 3       last_libindex = .current_libindex;
778 1502 3       last_libnumber = .current_libnumber;
779 1503 2
780 1504 2
781 1505 2 RETURN true;
782 1506 1 END;

```

! If help still not found
! Then go back to last library
! Set libindex
! Set libnumber
! Print help not available message

! If help found in new library
! Then clean up
! If last library not main library
! Then close that library
! If new library not main library

! Then change library name and
! indicate that there's more.

! Otherwise, simply reset it

! Reset last libindex
! Reset last libnumber

! Of search_libs

03FC 00000 SEARCH_LIBS:

57	59	0000V	CF	9E	00002	.	WORD	Save R2,R3,R4,R5,R6,R7,R8,R9		1363	
56	5E	FEEC	CE	9E	00007		MOVAB	CALL LBRHELP, R9			
			04	C1	0000C		MOVAB	-2767SP) SP			
	10	AC	08	C1	00011		ADDL3	#4, INDICES, R7		1414	
	10	AC	0C	AC	00016		ADDL3	#8, INDICES, R6		1415	
	53		12	A3	9E	0001A	MOVL	PRINTDATA, R3		1416	
	52			01	00	0001E	MOVAB	18(R3), R2			
	58			04	8A	00021	MOVL	#1, NOMSG		1428	
	62			AE	9E	00024	BICB2	#4, (R2)		1429	
0A	10	AE	14	AC	E8	00029	MOVAB	LIBRARYSTRING, LIBRARYNAME+4		1430	
	10	18		08	E1	0002D	BLBS	SUBPROMPTLEN, 1\$		1432	
	0000'	CF	04	BC	B5	00033	BBC	#11, HELP_FLAGS, 1\$		1433	
				05	13	00036	TSTW	KEYDESCS		1434	
				02	8A	00038	BEQL	1\$			
	62			03	11	0003B	BICB2	#2, (R2)		1435	
	62			02	88	0003D	BRB	2\$			
			04	AC	DD	00040	1\$: BISB2	#2, (R2)		1436	
				53	DD	00043	2\$: PUSHL	KEYDESCS			
			08	AC	DD	00045	PUSHL	R3		1439	
							PUSHL	OUTPUTWIDTH			

LBR_OUTPUTHELP V04=000	Prompting and library searching help function Routine search_libs	G 14 16-Sep-1984 02:04:00 14-Sep-1984 12:37:45	VAX-11 Bliss-32 V4.0-742 [LBR.SRC]OUTPUTHELP.B32;1	Page 23 (5)
				1438
06	69 54 2C 62 62	57 DD 00048 04 FB 0004A 50 D0 0004D 54 E9 00050 02 E1 00053 04 8A 00057 00D7 31 0005A 02 8A 0005D 02 E1 00060 66 D5 00066 18 19 00068 08 AE 10 04 AC 0006A AC DD 0006F 53 DD 00072 AC DD 00074 57 DD 00077 04 FB 00079 50 D0 0007C 54 E9 0007F 02 E1 00082 02 CE 00086 03 11 00089 01 CE 0008B 01 CE 0008E 02 EO 00091 5E DD 00095 AE 9F 00097 AE 9F 0009A 03 FB 0009D 50 D5 000A2 39 13 000A4 55 D6 000A6 55 D1 000A8 E4 13 000AB 58 DD 000AD AE 9F 000AF AE 9F 000B2 03 FB 000B5 50 E9 000BA 04 AC DD 000BD 53 DD 000CO AC DD 000C2 AE 9F 000C5 04 FB 000C8 50 D0 000CB 02 EO 000CE AE 9F 000D2 01 FB 000D5 54 E8 000DA 21 11 000DD 02 EO 000DF 67 D0 000E3 66 D0 000E7 02 88 000EA 04 AC DD 000ED 53 DD 000FO 08 AC DD 000F2	PUSHL R7 CALLS #4, CALL LBRHELP MOVL R0, STATOS BLBC STATUS 4\$ BBC #2, (R2), 3\$ BICB2 #4, (R2) BRW 16\$ BICB2 #2, (R2) BBC #2, HELP_FLAGS+1, 5\$ (R6) TSTL 5\$ BLSS 5\$ MOVL BINDICES, CURRENT_LIBINDEX PUSHL KEYDECS PUSHL R3 PUSHL OUTPUTWIDTH PUSHL R7 CALLS #4, CALL LBRHELP MOVL R0, STATOS BLBC STATUS 11\$ BBC #2, (R2), 6\$ MNEGL #2, CURRENT_LIBNUMBER BRB 7\$ MNEGL #1, CURRENT_LIBNUMBER MNEGL #1, USER_LIBNO BBS #2, (R2), 12\$ SP DEFLIB_ACMODE PUSHAB LIBRARYNAME CALLS #3, TRAN_NEXT_LIB TSTL R0 BEQL 10\$ INCL CURRENT_LIBNUMBER CMPL CURRENT_LIBNUMBER, (R6) BEQL 8\$ PUSHL NOMSG PUSHAB LIBRARYNAME PUSHAB CURRENT_LIBINDEX CALLS #3, OPEN_LIBRARY BLBC R0, 8\$ PUSHL KEYDECS R3 PUSHL OUTPUTWIDTH PUSHAB CURRENT_LIBINDEX CALLS #4, CALL LBRHELP MOVL R0, STATOS BBS #2, (R2), 9\$ PUSHAB CURRENT_LIBINDEX CALLS #1, CLOSE_LIBRARY BLBS STATUS, 8\$ BRB 11\$ BBS #2, (R2), 12\$ (R7), CURRENT_LIBINDEX MOVL (R6), CURRENT_LIBNUMBER BISB2 #2, (R2) PUSHL KEYDECS R3 PUSHL OUTPUTWIDTH	1442 1444 1445 1448 1450 1451 1453 1455 1454 1459 1460 1461 1463 1464 1465 1467 1468 1469 1472 1471 1473 1474 1475 1479 1481 1482 1483 1485
10	0000' CF	38:		
05	69 54 7E 62 55	48: 58: 68: 78: 88:		
6F	55 6E 62	88:		
	0000V CF	08 14		
	66			
	0000V CF D4	10 10 04 08 14		
08	69 54 62	10 10 03 50 53 08 14 04 02 AE 01 54 21 02 67 66 02 04 08		
21	08 AE 55 62	10\$: 10\$: 98: 10\$: 10\$:		

LBR
V04

LBR_OUTPUTHELP Prompting and library searching help function H 14
 V04=000 Routine search_libs 16-Sep-1984 02:04:00 14-Sep-1984 12:37:45 VAX-11 Bliss-32 V4.0-742
 [LBR.SRC]OUTPUTHELP.B32;1 Page 24 (5)

		57	DD 000F5	PUSHL	R7	: 1484
		04	FB 000F7	CALLS	#4. CALL LBRHELP	: 1
	54	50	DO 000FA	MOVL	R0, STATOS	: 1
	04	54	E8 000FD	BLBS	STATUS, 12\$: 1
	50	54	DO 00100	11\$: MOVL	STATUS, R0	: 1486
		04	00103	RET		
	66	55	D1 00104	12\$: CMPL	CURRENT_LIBNUMBER, (R6)	: 1489
		2B	13 00107	BEQL	16\$: 1
		66	D5 00109	TSTL	(R6)	: 1491
		07	19 0010B	BLSS	13\$: 1
	0000V CF	57	DD 0010D	PUSHL	R7	: 1492
		01	FB 0010F	CALLS	#1. CLOSE LIBRARY	: 1
		55	D5 00114	13\$: TSTL	CURRENT_LIBNUMBER	: 1493
		12	19 00116	BLSS	14\$: 1
	0000V CF	14	AC DD 00118	PUSHL	LIBNAME	: 1496
		10	AE 9F 0011B	PUSHAB	LIBRARYNAME	: 1
	0000' CF	02	FB 0011E	CALLS	#2. SWITCH_LIBNAME	: 1
		04	88 00123	BISB2	#4. PROMPT_FLAGS	: 1497
		03	11 00128	BRB	15\$: 1493
	67	14	BC B4 0012A	14\$: CLRW	@LIBNAME	: 1500
		08	AE DO 0012D	15\$: MOVL	CURRENT_LIBINDEX, (R7)	: 1501
	66	55	DO 00131	MOVL	CURRENT_LIBNUMBER, (R6)	: 1502
	50	01	DO 00134	16\$: MOVL	#1. R0	: 1505
		04	00137	RET		: 1506

; Routine Size: 312 bytes, Routine Base: \$CODES + 052E

LBR
V04

```

784      1507 1 XSBTTL 'Routine change lib';
785      1508 1 ROUTINE change_lib (getcmddesc, charpos, indices, libname) =
786      1509 2 BEGIN
787      1510 2
788      1511 2 ++ FUNCTIONAL DESCRIPTION:
789      1512 2
790      1513 2 This routine changes the library context currently in effect to
791      1514 2 the library specified by the command descriptor. It also removes
792      1515 2 the library specification from the command string.
793      1516 2
794      1517 2
795      1518 2 INPUTS:
796      1519 2
797      1520 2 getcmddesc = address of descriptor containing the new library name
798      1521 2
799      1522 2 charpos = pointer to "a" preceding library name
800      1523 2
801      1524 2 indices = address of data structure containing indices of
802      1525 2 libraries currently in use
803      1526 2
804      1527 2 libname = address of descriptor for the user default library name
805      1528 2
806      1529 2 OUTPUTS:
807      1530 2
808      1531 2 getcmddesc = library specification is removed from string
809      1532 2
810      1533 2 indices = if new library successfully found, updated to
811      1534 2 reflect library that was specified by the
812      1535 2 command string
813      1536 2
814      1537 2 libprompt = if new library successfully found, updated to
815      1538 2 the file name of that library
816      1539 2
817      1540 2 ROUTINE VALUE:
818      1541 2
819      1542 2 True, if new library found.
820      1543 2 False, if library unchanged.
821      1544 2
822      1545 2 MAP
823      1546 2 indices : REF BBLOCK,
824      1547 2 getcmddesc : REF BBLOCK;
825      1548 2
826      1549 2 BIND
827      1550 2 last_libindex = indices [hlISL_lastindex];
828      1551 2 last_libnumber = indices [hlISL_lastnumb];
829      1552 2
830      1553 2 LOCAL
831      1554 2 current_libindex,
832      1555 2 current_libnumber,
833      1556 2 deflib_acmode : BYTE,
834      1557 2 librarystring : BBLOCK [nam$C_maxrss],
835      1558 2 libraryname : BBLOCK [dsc$C_s_bln],
836      1559 2 name_end,
837      1560 2 name_len,
838      1561 2 nomsg,
839      1562 2 temp_end,
840      1563 2 user_libno;

                                ! Index of last library examined
                                ! No. of last library examined, relative to all defa
                                ! Index of library currently being searched
                                ! Number of library currently being searched
                                ! Logical name table number
                                ! Default library name string
                                ! String descriptor for library name
                                ! End of a string
                                ! Length of a string
                                ! Open library message flag
                                ! Location of first '/' in command string
                                ! HLP$LIBRARY number

```

```

841      1564 2 remove_terminator (.getcmddesc);
842      1565 2
843      1566 2
844      1567 2 IF (.charpos + 1) EQ
845          1568 3 (.getcmddesc [dsc$w_length] + .getcmddesc [dsc$sa_pointer])
846          1569 2     THEN RETURN false;
847      1570 2
848      1571 2 nomsg = true;
849      1572 2 libraryname [dsc$sa_pointer] = librarystring;
850      1573 2 make_upper_case (.getcmddesc, .getcmddesc [dsc$sa_pointer]);
851      1574 2
852      1575 2 name_len = .getcmddesc [dsc$w_length] +
853          1576 2     .getcmddesc [dsc$sa_pointer] - .charpos;
854      1577 2 IF (name_end = CH$IND (CH (.name_len, .charpos, %(' ')) EQ 0
855          1578 2     THEN name_end = .getcmddesc [dsc$w_length]
856          1579 2           + .getcmddesc [dsc$sa_pointer];
857      1580 2 IF (temp_end = CH$IND (CH (.name_len, .charpos, %('/')) NEQ 0
858          1581 2     THEN IF temp_end LSS name_end
859          1582 2         THEN name_end = temp_end;
860      1583 2 name_len = name_end - .charpos;
861      1584 2
862      1585 2 current_libnumber = -1;
863      1586 2 user_libno = -1;
864      1587 2 WHILE (tran_next_lib (libraryname, deflib_acmode, user_libno)
865          1588 2           EQ true)
866      DO BEGIN
867          LOCAL
868          filename : BBLOCK [dsc$C_s_bln],
869          filebuffer : VECTOR [filename_length, BYTE];
870
871          filename = 0;
872          filename [dsc$sa_pointer] = filebuffer;
873          switch_libname (libraryname, filename);
874
875          current_libnumber = .current_libnumber + 1;
876
877          IF CH$EQ (name_len - 1, filename [dsc$sa_pointer],
878              .name_len - 1, .charpos + 1, 0)
879
880              THEN IF (open_library (current_libindex, libraryname, .nomsg))
881                  THEN BEGIN
882                      IF .last_libnumber GEQ external_libnumber
883                          THEN close_library (last_libindex);
884                      switch libname (libraryname, .libname);
885                      last_libindex = .current_libindex;
886                      last_libnumber = .current_libnumber;
887                      getcmddesc [dsc$w_length] =
888                          .getcmddesc [dsc$w_length] - .name_len;
889                      getcmddesc [dsc$sa_pointer] =
890                          .getcmddesc [dsc$sa_pointer] + .name_len;
891                      RETURN true;
892
893                      END;
894
895                      END;
896
897                      ! If library file name was not found in logical name tables,

```

: F

```

898   1621 2 : then assume that the file name is actually a full file spec
899   1622 2 : for a library that exists but is not a user-defined default
900   1623 2 : library. Try to open that library.
901   1624 2 :
902   1625 2 :
903   1626 2 libraryname [dsc$w_length] = .name_len - 1;
904   1627 2 libraryname [dsc$w_pointer] = .charpos + 1;           ! Initialize library name
905   1628 2 :
906   1629 2 IF NOT (open_library (current_libindex, libraryname, .nomsg))    ! Try opening library
907   1630 2     THEN RETURN false;                                     ! If unsuccessful then give up and s
908   1631 2 :
909   1632 2 IF .last_libnumber GEQ external_libnumber
910   1633 2     THEN close_library (last_libIndex);
911   1634 2 switch libname (libraryname, .libname);          ! If last library not main Library
912   1635 2 last_libindex = .current_libindex;      Then close it
913   1636 2 last_libnumber = -1;                           Change library name
914   1637 2 getcmddesc [dsc$w_length] = .getcmddesc [dsc$w_length] - .name_len; Set libindex
915   1638 2 getcmddesc [dsc$w_pointer] = .getcmddesc [dsc$w_pointer] + .name_len; Set libnumber
916   1639 2 :
917   1640 2 RETURN true;                                Remove @ string from command
918   1641 1 END;                                         ! Return success
                                                       ! Of change_lib

```

OFFC 00000 CHANGE_LIB:										
5B	OC	SE	FEB	CE	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11		1508
5A	OC	AC		04	C1	00007	MOVAB	-324(SP), SP		1550
		AC		08	C1	0000C	ADDL3	#4, INDICES, R11		1551
		55	04	AC	D0	00011	ADDL3	#8, INDICES, R10		1565
				55	DD	00015	MOVL	GETCMDDESC, R5		
				01	FB	00017	PUSHL	R5		
	0000V	CF		08	AC	0001C	CALLS	#1, REMOVE TERMINATOR		1567
		56		51	01	A6	MOVAB	CHARPOS, R6		
		51		50	A6	9E	MOVAB	1(R6), R1		1568
		50		50	65	3C	MOVZWL	(R5), R0		
				04	A5	C0	ADDL2	4(R5), R0		
				50	D1	00027	CMPL	R1, R0		
					51	0002B	BNEQ	1S		
					03	12	BRW	11S		
					00F8	31	00030	1S:		
					01	00	MOVL	#1, NOMSG		1571
	40	AE		44	AE	00036	MOVAB	LIBRARYSTRING, LIBRARYNAME+4		1572
				04	A5	0003B	PUSHL	4(R5)		1573
					55	DD	PUSHL	R5		
	0000V	CF		02	FB	00040	CALLS	#2, MAKE_UPPER_CASE		1576
		53		65	3C	00045	MOVZWL	(R5), R3		
		53		04	A5	C0	ADDL2	4(R5), R3		
		54		56	C3	00048	SUBL3	R6, R3, NAME LEN		
				20	3A	00050	LOCC	#32, NAME_LEN, (R6)		1577
				02	12	00054	BNEQ	2S		
				51	D4	00056	CLRL	R1		
		52		51	D0	00058	2S:	1, NAME_END		
		52		03	12	0005B	BNEQ	3S		
		54		53	D0	0005D	MOVL	R3, NAME END		1579
	66			2F	3A	00060	LOCC	#47, NAME_LEN, (R6)		1580
				02	12	00064	BNEQ	4S		

LBR_OUTPUTHELP Prompting and library searching help function 16-Sep-1984 02:04:00 VAX-11 Bliss-32 V4.0-742
V04=000 Routine change_lib 14-Sep-1984 12:37:45 [LBR.SRC]OUTPUTHLP.B32;1

Page 28
(6)

LBR
V04

LBR.OUTPUTHELP Prompting and library searching help function M 14
V04-000 Routine change_lib 16-Sep-1984 02:04:00 VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1 Page 29
Page (6)

50 01 D0 00127 MOVL #1, R0
04 0012A RET
50 D4 0012B 11\$: CLRL R0
04 0012D RET

: 1640
: 1641
:

: Routine Size: 302 bytes, Routine Base: \$CODE\$ + 0666

```

920      1 XSBTTL 'Routine switch_libname';
921      1 ROUTINE switch_libname(newname, oldname) =
922      2 BEGIN
923
924      2 !++
925      2 !: FUNCTIONAL DESCRIPTION:
926
927      2 This routine inserts a new libname into the old descriptor.
928
929      2 !: INPUTS:
930
931      2 newname = address of a descriptor for the new library file name
932
933      2 oldname = address of a descriptor for the old library file name
934
935      2 !: OUTPUTS:
936
937      2 oldname = updated to specify the name of the new library
938
939      2 !: ROUTINE VALUE:
940
941      2 Always true.
942
943      2 !-- MAP
944
945      2 newname : REF BBLOCK,
946      2 oldname : REF BBLOCK;
947
948      2 LOCAL
949      2 rsabuf: BBLOCK [nam$c_maxrss],           ! buffer for resultant string
950      2 esabuf: BBLOCK [nam$c_maxrss],           ! buffer for expanded string
951      2 libfab: BBLOCK [fab$c_bln],             ! temporary FAB
952      2 libnam: BBLOCK [nam$c_bln];            ! temporary NAM block
953
954      P 1676 2 $NAM_INIT( NAM=libnam,
955      P 1677 2 RSA=rsabuf,
956      P 1678 2 RSS=nam$c_maxrss,
957      P 1679 2 ESA=esabuf,
958      P 1680 2 ESS=nam$c_maxrss);
959
960      P 1682 2 $FAB_INIT( FAB=libfab,
961      P 1683 2 FOP=NAM,
962      P 1684 2 FNA=.newname[dsc$a_pointer],
963      P 1685 2 FNS=.newname[dsc$w_length],
964      P 1686 2 NAM=libnam);
965
966      2 SPARSE(FAB=libfab);
967
968      2 oldname[dsc$w_length] = .libnam[nam$b_name];
969      2 CHSMOVE(.libnam[nam$b_name], .libnam[nam$l_name], .oldname[dsc$a_pointer]);
970
971      2 RETURN true;
972      1 END;                                     ! Of switch_libname

```

.EXTRN SYSPARSE

003C 00000 SWITCH_LIBNAME:									
							WORD	Save R2, R3, R4, R5	1643
0060	BF	00	5E	FD50	CE	9E 00002	MOVAB	-688(SP), SP	
			6E		00	2C 00007	MOVCS	#0, (SP), #0, #96, SRMS_PTR	1680
			6E	6002	8F	80 0000F	MOVW	#24578, SRMS_PTR	
		02	AE		01	8E 00014	MNEG B	#1, SRMS_PTR+2	
		04	AE	FF00	CD	9E 00018	MOVAB	RSABUF, SRMS_PTR+4	
		0A	AE		01	8E 0001E	MNEG B	#1, SRMS_PTR+10	
0050	8F	00	OC	00B0	CE	9E 00022	MOVAB	ESABUF, SRMS_PTR+12	
			6E		00	2C 00028	MOVCS	#0, (SP), #0, #80, SRMS_PTR	1686
			60		AE	0002F	MOVW	#20483, SRMS_PTR	
		60	AE	5003	8F	80 00031	MOVL	#16777216, SRMS_PTR+4	
		64	AE	01000000	8F	00 00037	MOVB	#2, SRMS_PTR+22	
		76	AE		02	90 0003F	MOV B	#2, SRMS_PTR+31	
		7F	AE		02	90 00043	MOVAB	LIBNAM, SRMS_PTR+40	
		0088	CE		6E	9E 00047	MOVL	NEWNAME, R0	
		008C	CE		04	AC 0004C	MOVL	4(R0), SRMS_PTR+44	
		0094	CE		04	A0 00050	MOVB	(R0), SRMS_PTR+52	
		00000000G	00		60	AE 9F 0005B	PUSHAB	LIBFAB	1688
			50		01	FB 0005E	CALLS	#1, SYSSPARSE	
			50		08	AC 00065	MOVL	OLDNAME, R0	1690
			60		3B	AE 9B 00069	MOVZBW	LIBNAM+59, (R0)	
			51		3B	AE 9A 0006D	MOVZBL	LIBNAM+59, R1	1691
04	B0	4C	BE		51	28 00071	MOVCS	R1, ALIBNAM+76, A4(R0)	
			50		01	00 00077	MOVL	#1, R0	1693
					04	0007A	RET		1694

; Routine Size: 123 bytes, Routine Base: SCODE\$ + 0794


```

1031      1752 2 deflibdesc [dsc$w_length] = maxliblen;
1032      1753 2 deflibdesc [dsc$w_pointer] = deflibnam;
1033      1754 2
1034      1755 2 IF ..userlibno EQL 0
1035      1756 3 THEN BEGIN
1036          1757 3     deflibdesc [dsc$w_length] = .hlplibnam [0];
1037          1758 3     CHSMOVE (.deflibdesc [dsc$w_length], hlplibnam + 1, deflibnam);
1038          1759 3 END
1039      1760 2
1040      1761 2 ELSE BEGIN
1041          1762 3     deflibdesc [dsc$w_length] = .libnamefao [0];
1042          1763 3     CHSMOVE (.deflibdesc [dsc$w_length], libnamefao + 1, deflibnam);
1043          1764 4     IF NOT $FAO (deflibdesc, deflibdesc, deflibdesc, ..userlibno)
1044              1765 3         THEN RETURN false;
1045          1766 2 END;
1046      1767 2
1047      1768 2 IF .help_flags [..deflibacmode + 1] NEQ 0
1048          1769 3 THEN BEGIN
1049              1770 3     deflibdsbmsk = XX'00060503';
1050              1771 3     libname [dsc$w_length] = namSc_maxrss;
1051              P 1772 3     status = STRNL0G (LOGNAM = def[ibdesc,
1052                  P 1773 3             DSBMSK = .deflibdsbmsk [..deflibacmode],
1053                  P 1774 3             RSLLEN = .libname,
1054                  P 1775 3             RSLBUF = .libname);
1055              1776 4     IF .status AND (.status NEQ SSS_NOTRAN)
1056                  1777 4         THEN BEGIN
1057                      1778 4             .userlibno = ..userlibno + 1;
1058                      1779 4             RETURN true;
1059                      1780 3         END;
1060                  1781 2     END;
1061      1782 2
1062      1783 2
1063      1784 2     ! If current logical name table not enabled or logical name unsuccessfully
1064          1785 2     translated, then recursively call this routine to search next table.
1065      1786 2
1066      1787 2
1067      1788 2     IF (.deflibacmode = ..deflibacmode + 1) GTR 2
1068          1789 2         THEN RETURN false;
1069          1790 2     .userlibno = 0;
1070          1791 2     RETURN tran_next_lib (.libname, .deflibacmode, .userlibno);
1071          1792 2
1072      1793 1 END;

```

! Initialize logical name descriptor
! If first logical name in table
! Then special case, do not use FAO
! Else use FAO to combine logical na
! If searching of current logical na
! Then translate logical name for th
! If logical name is successfully tr
! Then increment lib no. for next se
! And return true
! Increment access mode
! If out of tables, then return fals
! Reset lib no. to start of table
! Make recursive call
! Of tran_next_lib

.PSECT S0WNS,NOEXE,2

	0B 00038 HLPLIBNAME:	
59 52 41 52 42 49 4C 24 50 4C 48 00039	:BYTE 11	
OF 00044 LIBNAMEFAO:	:ASCII \HPLSLIBRARY\	
	:BYTE 15	
	:ASCII \HPLSLIBRARY_!UW\	
	MAXLIBLEN= 15	
	.EXTRN SYSSFAO, SYSSTRNLOG	

.PSECT \$CODE\$,NOWRT,2

00FC 00000 TRAN_NEXT LIB:									
							WORD	Save R2,R3,R4,R5,R6,R7	: 1696
							MOVAB	HLPLIBNAM, R7	
							SUBL2	#28, SP	
							MOVL	USERLIBNO, R6	
							TSTL	(R6)	
							BGEQ	1\$	
							CLRL	2DEFLIBACMODE	
							CLRL	(R6)	
							MOVW	#15, DEFLIBDESC	
							MOVAB	DEFLIBNAM, DEFLIBDESC+4	
							TSTL	(R6)	
							BNEQ	2\$	
							MOVZBW	HLPLIBNAM, DEFLIBDESC	
							MOVC3	DEFLIBDESC, HLPLIBNAM+1, DEFLIBNAM	
							BRB	3\$	
							MOVZBW	LIBNAMEFAO, DEFLIBDESC	
							MOVC3	DEFLIBDESC, LIBNAMEFAO+1, DEFLIBNAM	
							PUSHL	(R6)	
							DEFLIBDESC		
							PUSHAB	DEFLIBDESC	
							PUSHAB	DEFLIBDESC	
							PUSHAB	DEFLIBDESC	
							CALLS	#4, SYSSFAO	
							BLBC	R0, 5\$	
							MOVL	2DEFLIBACMODE, R0	
							MOVAB	1(R0), R2	
							EXTZV	R2, #1, HELP_FLAGS, R1	
							BEQL	4\$	
							MOVL	#394499, DEFLIBDSBMSK	
							MOVZBW	#255, ALIBNAME	
							MOVZBL	DEFLIBDSBMSK[R0], -(SP)	
							CLRQ	- (SP)	
							PUSHL	LIBNAME	
							PUSHL	LIBNAME	
							PUSHAB	DEFLIBDESC	
							CALLS	#6, SYSTRNLOG	
							BLBC	STATUS, 4\$	
							CMPL	STATUS, #1577	
							BEQL	4\$	
							INCL	(R6)	
							MOVL	#1, R0	
							RET		
							ADDL3	#1, 2DEFLIBACMODE, R0	
							MOVL	R0, 2DEFLIBACMODE	
							CMPL	R0, #2	
							BGTR	5\$	
							CLRL	(R6)	
							PUSHL	R6	
							MOVQ	LIBNAME, -(SP)	
							CALLS	#3, TRAN_NEXT_LIB	
							RET		
							CLRL	R0	
							RET		

: Routine Size: 181 bytes. Routine Base: \$CODE\$ + 080F

LBR_OUTPUTHELP Prompting and library searching help function
V04-000 Routine tran_next_lib

F 15
16-Sep-1984 02:04:00
14-Sep-1984 12:37:45

VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1

Page 35
(8)

LB
VO

```
1074 1794 1 %SBTTL 'Routine open_library';
1075 1795 1 ROUTINE open_library (libindex, libname, nomsgflag) =
1076 1796 2 BEGIN
1077 1797 2
1078 1798 2 ++
1079 1799 2 ! FUNCTIONAL DESCRIPTION:
1080 1800 2
1081 1801 2 Open the library with the specified name and return its index.
1082 1802 2
1083 1803 2 ! INPUTS:
1084 1804 2
1085 1805 2 libindex = address of longword to contain index of opened library
1086 1806 2
1087 1807 2 libname = address of descriptor for library name
1088 1808 2
1089 1809 2 nomsgflag = flag that is true if open errors should not be signalled
1090 1810 2
1091 1811 2 ! OUTPUTS:
1092 1812 2
1093 1813 2 libindex : as described above
1094 1814 2
1095 1815 2 ! ROUTINE VALUE:
1096 1816 2
1097 1817 2 True, if library successfully opened.
1098 1818 2 False, if unsuccessful.
1099 1819 2
1100 1820 2 --
1101 1821 2
1102 1822 2 ! MAP
1103 1823 2 libname : REF BBLOCK;
1104 1824 2
1105 1825 2 ! EXTERNAL
1106 1826 2 lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL); ! RMS STV from librarian
1107 1827 2
1108 1828 2 ! LOCAL
1109 1829 2 filnamdesc : BBLOCK [dsc$c_s_bln],
1110 1830 2 help_defname : BBLOCK [dsc$c_s_bln],
1111 1831 2 helpfilename : BBLOCK [nam$c_maxrss],
1112 1832 2 help_func,
1113 1833 2 help_type,
1114 1834 2 namblk : BBLOCK [nam$c_bln],
1115 1835 2 status; ! Library name descriptor
1116 1836 2
1117 P 1837 2 $NAM_INIT (NAM = namblk, ! Initialize the NAM block
1118 P 1838 2 RSS = nam$c_maxrss,
1119 P 1839 2 RSA = helpfilename,
1120 P 1840 2 ESS = nam$c_maxrss,
1121 P 1841 2 ESA = helpfilename);
1122 1842 2
1123 1843 2 help_func = lbr$g_read; ! Will be reading the library
1124 1844 2 help_type = lbr$g_typ_hlp; ! Library is of help type
1125 1845 2
1126 1846 2 !
1127 1847 2 ! Call librarian to initialize control. Stop if error.
1128 1848 2
1129 1849 2
1130 1850 3 IF NOT (status = lbr$ini_control (.libindex, help_func, help_type, namblk))
```

```

1131 1851 2 THEN
1132 1852 2 BEGIN
1133 1853 2 SIGNAL (.status);
1134 1854 2 RETURN .status OR stssm_inhib_msg;
1135 1855 2 END;
1136 1856 2
1137 1857 2 : Call Librarian to open library. Stop if error should be flagged.
1138 1858 2
1139 1859 2
1140 1860 2
1141 1861 2 help_defname [dsc$w_length] = .syshelp [0];           ! Initialize default library directory and type
1142 1862 2 help_defname [dsc$w_pointer] = syshelp [1];
1143 1863 2
1144 1864 3 IF NOT (status = lbr$open (.libindex, .libname, 0, help_defname))
1145 1865 3 AND NOT .nomsgflag
1146 1866 3 THEN BEGIN
1147 1867 3     IF (filnamdesc [dsc$w_length] = .namblk [nam$b_esl]) NEQ 0
1148 1868 3         THEN filnamdesc [dsc$w_pointer] = .namblk [nam$l_esa]
1149 1869 4     ELSE BEGIN
1150 1870 4         filnamdesc [dsc$w_length] = .libname [dsc$w_length];
1151 1871 4         filnamdesc [dsc$w_pointer] = .libname [dsc$w_pointer];
1152 1872 3     END;
1153 1873 3     SIGNAL (shrs_openin OR hlp$e_facility OR stssk_error,
1154 1874 3             1, filnamdesc, .status, .lbr$gl_rmsstv);
1155 1875 3     status = shrs_openin OR hlp$e_facility OR stssk_error OR stssm_inhib_msg;
1156 1876 2
1157 1877 2
1158 1878 2 RETURN .status;
1159 1879 2
1160 1880 1 END;                                              !Of open_library

```

.EXTRN LBR\$GL_RMSSTV

007C 00000 OPEN_LIBRARY:							
0060	BF	00	56 0000000G	00 9E 00002	.WORD	Save R2,R3,R4,R5,R6	: 1795
			5E FE88	CE 9E 00009	MOVAB	LIB\$SIGNAL, R6	
			6E 08	00 2C 0000E	MOVCS	-376(SP), SP	
				AE 00015		#0, (SP), #0, #96, SRMS_PTR	1841
		08	AE 6002	8F B0 00017	MOVW	#24578, SRMS_PTR	
		0A	AE AE	01 8E 0001D	MNEG	#1, SRMS_PTR+2	
		0C	AE 68	AE 9E 00021	MOVAB	HELPFILENAME, \$RMS_PTR+4	
		12	AE 68	01 8E 00026	MNEG	#1, SRMS_PTR+10	
		14	AE 68	AE 9E 0002A	MOVAB	HELPFILENAME, \$RMS_PTR+12	
		04	AE 6E	01 D0 0002F	MOVL	#1, HELP_FUNC	1843
				03 D0 00033	MOVL	#3, HELP_TYPE	1844
				08 AE 9F 00036	PUSHAB	NAMBLK	1850
				04 AE 9F 00039	PUSHAB	HELP_TYPE	
				0C AE 9F 0003C	PUSHAB	HELP_FUNC	
				04 AC DD 0003F	PUSHL	LIBINDEX	
		0000000G	00	04 FB 00042	CALLS	#4, LBR\$INI_CONTROL	
		53	50	D0 00049	MOVL	R0, STATUS	
		0E	53 E8	0004C	BLBS	STATUS, 1\$	
			53 DD	0004F	PUSHL	STATUS	
			66	01 FB 00051	CALLS	#1, LIB\$SIGNAL	1853

LBR_OUTPUTHELP Prompting and library searching help function
V04=000 Routine open_library

I 15

16-Sep-1984 02:04:00
14-Sep-1984 12:37:45

VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1

Page 38
(9)

LB
VO

50	S3	10000000	8F	C9	00054		BISL3	#268435456, STATUS, R0	: 1854
				04	0005C		RET		
F0	AD	0000'	CF	9B	0005D	1\$:	MOVZBW	SYSHELP, HELP_DEFNAME	: 1861
F4	AD	0000'	CF	9E	00063		MOVAB	SYSHELP+1, HELP_DEFNAME+4	: 1862
		F0	AD	9F	00069		PUSHAB	HELP_DEFNAME	: 1864
				7E	D4	0006C	CLRL	- (SPT)	
52		08	AC	DD	0006E		MOVL	LIBNAME, R2	
			52	DD	00072		PUSHL	R2	
		04	AC	DD	00074		PUSHL	LIBINDEX	
00000000G	00		04	FB	00077		CALLS	#4, LBR\$OPEN	
53			50	DO	0007E		MOVL	RO, STATUS	
38			53	E8	00081		BLBS	STATUS, 4S	
34		OC	AC	E8	00084		BLBS	NOMSGFLAG, 4S	: 1865
F8	AD	13	AE	9B	00088		MOVZBW	NAMBLK+11, FILNAMDESC	: 1867
FC	AD	14	AE	DO	0008F		BEQL	2\$: 1868
F8	AD		07	13	0008D		MOVL	NAMBLK+12, FILNAMDESC+4	
FC	AD		09	11	00094		BRB	3\$	
			62	BO	00096	2\$:	MOVW	(R2), FILNAMDESC	: 1870
		04	A2	DO	0009A		MOVL	4(R2), FILNAMDESC+4	: 1871
00000000G	00		00	DD	0009F	3\$:	PUSHL	LBRSGL_RMSSTV	: 1874
			53	DD	000A5		PUSHL	STATUS	
		F8	AD	9F	000A7		PUSHAB	FILNAMDESC	: 1873
0076109A		01	DD	000AA			PUSHL	#1	
66			8F	DD	000AC		CALLS	#7737498	
53	1076109A	05	FB	000B2			MOVL	#5 LIB\$SIGNAL	: 1875
50			8F	DO	000B5		MOVL	#276172954, STATUS	: 1878
		53	DO	000BC		4\$:	RET	STATUS, RO	: 1880
			04	000BF					

; Routine Size: 192 bytes. Routine Base: \$CODE\$ + 08C4

```

1162    1881 1 %SBTTL 'Routine close_library';
1163    1882 1 ROUTINE close_library(libindex) =
1164    1883 2 BEGIN
1165    1884 2
1166    1885 2++ FUNCTIONAL DESCRIPTION:
1167    1886 2
1168    1887 2 Close the open help library.
1169    1888 2
1170    1889 2
1171    1890 2 INPUTS:
1172    1891 2
1173    1892 2 libindex = address of longword containing index of library to close.
1174    1893 2
1175    1894 2 OUTPUTS:
1176    1895 2
1177    1896 2 None.
1178    1897 2
1179    1898 2 ROUTINE VALUE:
1180    1899 2
1181    1900 2 Always true.
1182    1901 2
1183    1902 2---
1184    1903 2
1185    1904 2 LOCAL
1186    1905 2 status;
1187    1906 2
1188    1907 3 IF NOT (status = lbr$close (.libindex))
1189    1908 2 THEN SIGNAL (.status);
1190    1909 2
1191    1910 2 RETURN true
1192    1911 2
1193    1912 1 END;                                !Of close_library

```

0000 0000 CLOSE_LIBRARY:							
					.WORD	Save nothing	: 1882
00000000G	00	04	AC DD 00002	PUSHL	LIBINDEX		: 1907
	09		01 FB 00005	CALLS	#1, LBR\$CLOSE		
00000000G	00		50 E8 0000C	BLBS	STATUS, 1\$: 1908
	50		50 DD 0000F	PUSHL	STATUS		
			01 FB 00011	CALLS	#1, LIB\$SIGNAL		: 1910
			01 D0 00018 1\$:	MOVL	#1, R0		
			04 00018	RET			: 1912

: Routine Size: 28 bytes. Routine Base: \$CODE\$ + 0984

```

1195 1913 1 XSBTTL 'Routine setup_keys';
1196 1914 1 ROUTINE setup_keys (getcmddesc, keydescs, truekeys) =
1197 1915 2 BEGIN
1198 1916 2
1199 1917 2
1200 1918 2
1201 1919 2
1202 1920 2
1203 1921 2
1204 1922 2
1205 1923 2
1206 1924 2
1207 1925 2
1208 1926 2
1209 1927 2
1210 1928 2
1211 1929 2
1212 1930 2
1213 1931 2
1214 1932 2
1215 1933 2
1216 1934 2
1217 1935 2
1218 1936 2
1219 1937 2
1220 1938 2
1221 1939 2
1222 1940 2
1223 1941 2
1224 1942 2
1225 1943 2
1226 1944 2
1227 1945 2
1228 1946 2
1229 1947 2
1230 1948 2
1231 1949 2
1232 1950 2
1233 1951 2
1234 1952 2
1235 1953 2
1236 1954 2
1237 1955 2
1238 1956 2
1239 1957 2
1240 1958 2
1241 1959 2
1242 1960 2
1243 1961 2
1244 1962 2
1245 1963 2
1246 1964 2
1247 1965 2
1248 1966 2
1249 1967 2
1250 1968 2
1251 1969 2

1 XSBTTL 'Routine setup_keys';
1 ROUTINE setup_keys (getcmddesc, keydescs, truekeys) =
2 BEGIN
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967

```

```

1252      1970 5   IF (parm_begin = CH$IND_NOT_CH (.line_end - .parm_end,
1253          1971 4       .parm_end, %ASCII ' ') EQL 0
1254          1972 4   THEN parm_begin = .line_end;
1255          1973 4
1256          1974 4   IF .next_qual NEQ 0
1257          1975 4       AND .parm_begin GEQU .next_qual
1258          1976 5   THEN BEGIN
1259          1977 5       parm_begin = .next_qual;
1260          1978 5       next_qual = CH$IND_CH (.line_end - .parm_begin - 1,
1261          1979 5           .parm_begin + 1, %ASCII '/');
1262          1980 4   END;
1263          1981 4
1264          1982 5   IF (parm_end = CH$IND_CH (.line_end - .parm_begin,
1265          1983 4       .parm_begin, %ASCII ') ) EQL 0
1266          1984 4   THEN parm_end = .line_end;
1267          1985 4
1268          1986 4   IF .next_qual NEQ 0
1269          1987 4       AND .parm_end GTRU .next_qual
1270          1988 4   THEN parm_end = .next_qual;
1271          1989 4
1272          1990 4   END
1273          1991 4
1274          1992 4   UNTIL (.parm_end - .parm_begin EQL 0
1275          1993 3       OR CH$NEQ (1, .parm_begin, 1, paren));
1276          1994 3
1277          1995 3
1278          1996 3   ! Fill in string descriptor
1279          1997 3
1280          1998 3
1281          1999 4   BEGIN
1282          2000 4
1283          2001 4   BIND
1284          2002 4       curkeydesc = keydescs [dsc$c_s_bln * .level] : BBLOCK;    ! Descriptro for current key
1285          2003 4
1286          2004 4   IF (curkeydesc [dsc$w_length] = .parm_end - .parm_begin) EQL 0 ! If key length is zero
1287          2005 5   THEN BEGIN
1288          2006 5       IF .level EQL 0
1289          2007 5           THEN prompt_flags = .prompt_flags OR hcf$m_stay;
1290          2008 5           truekeys [0] = .level;
1291          2009 5           EXITLOOP;
1292          2010 4
1293          2011 4       curkeydesc [dsc$sa_pointer] = .parm_begin;
1294          2012 3
1295          2013 2   END;
1296          2014 2
1297          2015 2   RETURN true
1298          2016 2
1299          2017 1 END;                                         !Of setup_keys

```

LBR_OUTPUTHELP Prompting and library searching help function
V04=000 Routine setup_keys

M 15
16-Sep-1984 02:04:00
14-Sep-1984 12:37:45

VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1

Page 42
(11)

LB
VO

			08	BC	0000C		MOVL	GETCMDDESC, R3	1960
			04	AC	0000E		PUSHL	4(R3)	
			04	A3	00012		PUSHL	R3	
			53	DD	00015		CALLS	#2, MAKE_UPPER_CASE	
		0000V	CF	02	FB	00017	MOVL	4(R3), PARM_END	1961
			55	04	A3	0001C	MOVZWL	(R3), LINE_END	1962
			52	63	3C	00020	ADDL2	4(R3), LINE_END	
			52	04	A3	00023	LOCC	#47, (R3), #4(R3)	1963
04	B3		63	2F	3A	00027	BNEQ	1S	
				02	12	0002C	CLRL	R1	
			56	51	D4	0002E	MOVL	R1, NEXT_QUAL	
				51	D0	00030	CLRL	LEVEL	1970
50		52	52	54	D4	00033	SUBL3	PARM-END, LINE-END, R0	
65		50	50	20	3B	00039	SKPC	#32, R0, (PARM-END)	
				02	12	0003D	BNEQ	3S	
			53	51	D4	0003F	CLRL	R1	
				03	D0	00041	MOVL	R1, PARM_BEGIN	1971
			53	52	D0	00044	BNEQ	4S	
				52	D0	00046	MOVL	LINE_END, PARM_BEGIN	1972
			53	56	D5	00049	TSTL	NEXT_QUAL	1974
				1A	13	0004B	BEQL	6S	
			56	53	D1	0004D	CMPL	PARM_BEGIN, NEXT_QUAL	1975
				15	1F	00050	BLSSU	6S	
50		53	56	D0	00052	MOVL	NEXT_QUAL, PARM_BEGIN	1977	
		52	53	C3	00055	SUBL3	PARM_BEGIN, LINE_END, R0	1978	
01	A3	50	50	D7	00059	DECL	R0		
				2F	3A	0005B	LOCC	#47, R0, 1(PARM_BEGIN)	
			50	02	12	00060	BNEQ	5S	
				51	D4	00062	CLRL	R1	
50		56	51	D0	00064	5S:	MOVL	R1, NEXT_QUAL	
63		52	53	C3	00067	6S:	SUBL3	PARM_BEGIN, LINE_END, R0	1982
		50	20	3A	0006B	LOCC	#32, R0, (PARM_BEGIN)		
			02	12	0006F	BNEQ	7S		
			51	D4	00071	CLRL	R1		
		55	51	D0	00073	7S:	MOVL	R1, PARM_END	
				03	12	00076	BNEQ	8S	
		55	52	D0	00078	MOVL	LINE_END, PARM_END	1983	
				56	D5	0007B	TSTL	NEXT_QUAL	1984
		55	08	13	0007D	BEQL	9S	1986	
			56	D1	0007F	CMPL	PARM_END, NEXT_QUAL	1987	
		56	03	1B	00082	BLEQU	9S		
			55	D0	00084	MOVL	NEXT_QUAL, PARM END	1988	
		53	55	D1	00087	9S:	CMPL	PARM_END, PARM_BEGIN	1992
			05	13	0008A	BEQL	10S		
		57	63	91	0008C	CMPB	(PARM_BEGIN), PAREN	1993	
			A4	13	0008F	BEQL	2S		
51		50	08	BC	44	7E	00091	10S:	KEYDESCS[LEVEL], R0
			55	C3	00096		SUBL3	PARM BEGIN, PARM_END, R1	2002
			60	B0	0009A		MOVW	R1, (R0)	2004
				51	D5	0009D	TSTL	R1	
				0F	12	0009F	BNEQ	12S	
				54	D5	000A1	TSTL	LEVEL	2006
		0000'	05	12	000A3		BNEQ	11S	
		OC	02	88	000A5		BISB2	#2, PROMPT FLAGS	2007
		BC	54	90	000AA	11S:	MOVB	LEVEL, ATROEKEYS	2008
			OE	11	000AE		BRB	13S	2005

LBR_OUTPUTHELP Prompting and library searching help function N 15
V04=000 Routine setup_keys 16-Sep-1984 02:04:00 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:37:45 [LBR.SRC]OUTPUTHELP.B32;1 Page 43
(11)

04 A0	53 D0 000B0 12\$:	MOVL PARM BEGIN, 4(R0)	: 2011
09	54 D6 000B4	INCL LEVEC	: 1966
	54 D1 000B6	CMPL LEVEL, #9	:
	03 1A 000B9	BGTRU 13\$:
50	FF77 31 000BB	BRW 2\$:
	01 D0 000BE 13\$:	MOVL #1, R0	: 2015
	04 000C1	RET	: 2017

: Routine Size: 194 bytes, Routine Base: \$CODE\$ + 09A0

BUDERFGHIIJKLNUOWFQHIIJKLNENBQWFGHIIJKLNENBQWFGHII

```
1301      2018 1 %SBTTL 'Routine call_lbrhelp';
1302      2019 1 ROUTINE call_lbrhelp(helpplibindex, outputwidth, printdata, keydescs) =
1303          2 BEGIN
1304          2
1305          2      /**
1306          2      * FUNCTIONAL DESCRIPTION:
1307          2
1308          2      This routine calls the Librarian function to extract help from
1309          2      a particular help library.
1310
1311          2      INPUTS:
1312
1313          2          helpplibindex = address of longword containing help library index
1314          2          outputwidth = address of longword containing width of output line
1315          2          printdata = address of data structure containing information for
1316          2          the output driver
1317          2          keydescs = address of vector of individual descriptors for each keyword
1318
1319
1320          2      OUTPUTS:
1321
1322          2          None.
1323
1324          2      ROUTINE VALUE:
1325
1326          2          Status of call for help.
1327
1328          2
1329          2
1330          2
1331          2
1332          2
1333          2
1334          2
1335          2
1336          2
1337          2
1338          2
1339          2
1340          2
1341          2
1342          2
1343          2
1344          2
1345          2
1346          2
1347          2
1348          2
1349          2
1350          2
1351          2
1352          2
1353          2
1354          2
1355          2
1356          2
1357          2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2047 2
1302      2048 2
1303      2049 2
1304      2050 2
1305      2051 2
1306      2052 2
1307      2053 2
1308      2054 2
1309      2055 2
1310      2056 2
1311      2057 2
1312      2058 2
1313      2059 2
1314      2060 2
1315      2061 2
1316      2062 2
1317      2063 2
1318      2064 2
1319      2065 2
1320      2066 2
1321      2067 2
1322      2068 2
1323      2069 2
1324      2070 2
1325      2071 2
1326      2072 2
1327      2073 2
1328      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311      2059 2
1312      2060 2
1313      2061 2
1314      2062 2
1315      2063 2
1316      2064 2
1317      2065 2
1318      2066 2
1319      2067 2
1320      2068 2
1321      2069 2
1322      2070 2
1323      2071 2
1324      2072 2
1325      2073 2
1326      2074 2
1301      2049 2
1302      2050 2
1303      2051 2
1304      2052 2
1305      2053 2
1306      2054 2
1307      2055 2
1308      2056 2
1309      2057 2
1310      2058 2
1311
```

```

: 1358    2075 2 IF .status EQL lbr$_endtopic THEN      ! Abort help on this topic?
: 1359    2076 2 BEGIN                                     ! Set flag to abort additional library listing and
: 1360    2077 2 end_topic_flag = true;                  ! change error to true
: 1361    2078 2 status = true;
: 1362    2079 2 END;
: 1363    2080 2
: 1364    2081 2 RETURN .status
: 1365    2082 1 END;

```

! Of call_lbrhelp

.EXTRN LBR\$GL_CONTROL

					0004 00000 CALL_LBRHELP:			
					.WORD	Save R2		2019
					MOVL	LBR\$GL_CONTROL, R0		2059
					MOVL	14(R0), R2		2065
					BISB2	#1, 5(R2)		2072
					MOVL	KEYDESCS, R0		2071
					PUSHAB	72(R0)		2070
					PUSHAB	64(R0)		2069
					PUSHAB	56(R0)		2068
					PUSHAB	48(R0)		2072
					PUSHAB	40(R0)		2071
					PUSHAB	32(R0)		2070
					PUSHAB	24(R0)		2069
					PUSHAB	16(R0)		2068
					PUSHAB	8(R0)		2072
					PUSHL	R0		2071
					PUSHL	PRINTDATA		2066
					PUSHAB	OUTPUT DRIVER		2072
					MOVQ	HEPLIBINDEX, -(SP)		2073
					CALLS	#14, LBR\$GET_HELP		2075
					BICB2	#1, 5(R2)		2077
					CMPL	STATUS, #LBR\$_ENDTOPIC		2078
					BNEQ	1\$		2082
					MOVBL	#1, END_TOPIC_FLAG		
					MOVL	#1, STATUS		
					RET			

; Routine Size: 88 bytes, Routine Base: \$CODES + 0A62

```

1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249

```

```

1424      2     status;
1425
1426      2     status = true;
1427      2     flags = ..helpflags;
1428
1429      2     IF NOT .print_flags [hpdsV_all] AND
1430          ((.flags AND hlpSm_nohtxt) NEQ 0)
1431      3     THEN BEGIN
1432          print_flags [hpdsV_found] = false;
1433          RETURN true;
1434          END;
1435
1436      2     print_flags [hpdsV_found] = true;
1437      2     print_flags [hpdsV_all] = true;
1438
1439      2     keylevel = ..helplevel;
1440      2     output_desc = 0;
1441      2     output_desc [dsc$w_length] = .linedesc [dsc$w_length];
1442      2     output_desc [dsc$w_pointer] = output_buf;
1443
1444      2     IF .flags EQL 0
1445          THEN prompt_flags = .prompt_flags OR hcfSm_info;
1446
1447      2     !
1448      2     ! If processing key name and moving down a prompt level,
1449      2     ! then build subtopic prompt
1450
1451
1452      2     IF ((.flags AND hlpSm_keynamlin) NEQ 0) AND
1453          ((.prompt_flags AND hcfSm_stay) EQL 0)
1454      3     THEN BEGIN
1455          IF (.prompt_flags AND hcfSm_info) NEQ 0
1456              THEN prompt_flags = .prompt_flags OR hcfSm_stay
1457          ELSE BEGIN
1458              IF (.keylevel NEQ .help_level) AND (.output_desc [dsc$w_length] NEQ 0)
1459                  THEN BEGIN
1460                      help_level = .keylevel;
1461                      CHSMOVE (.output_desc [dsc$w_length], .linedesc [dsc$w_pointer],
1462                               .sub_prompt_ptr);
1463                      length_array [.help_level] = .output_desc [dsc$w_length];
1464                      sub_prompt_ptr = .sub_prompt_ptr + .output_desc [dsc$w_length];
1465                      (.sub_prompt_ptr)<0,85 = 32;
1466                      sub_prompt_ptr = .sub_prompt_ptr + 1;
1467                  END;
1468
1469                  END;
1470                  add_info_level = .keylevel;
1471
1472
1473      2     ! If no help text found, then stay at current level and signal that
1474      2     ! help was not found.
1475
1476
1477      2     IF (.flags AND hlpSm_nohtxt) NEQ 0
1478          THEN BEGIN
1479              prompt_flags = .prompt_flags OR hcfSm_stay;
1480              IF (.print_flags [hpdsV_init] EQL 0)

```

```

1481      2197 3      THEN print_flags [hpdsV_init] = 1;
1482      2198 2      END;
1483      2199 2
1484      2200 2
1485      2201 2      ! If additional info to be found, say so
1486      2202 2
1487      2203 2
1488      2204 2      IF (.flags AND hlp$M_otherinfo) NEQ 0
1489      2205 2          THEN prompt_flags = .prompt_flags OR hcf$M_more;
1490      2206 2
1491      2207 2      !
1492      2208 2      Format output line and then call user supplied output routine.
1493      2209 2
1494      2210 2
1495      2211 3      IF ((.true_keys NEQ 0) OR
1496      2212 3          .help_flags [hlp$V_help] OR
1497      2213 3          ((.flags AND hlp$M_otherinfo) NEQ 0))           ! Not at first topic level
1498      2214 2          THEN IF (.keylevel GTR 0) OR
1499      2215 3              ((.flags AND hlp$M_keynamlin) EQL 0)           ! Want help on help with topic list
1500      2216 3          THEN BEGIN
1501      2217 3              IF .linedesc [dsc$W_length] NEQ 0
1502      2218 4                  THEN BEGIN
1503      2219 4
1504      2220 5                  IF ((.flags AND hlp$M_keynamlin) NEQ 0)
1505      2221 4                      THEN spaces = (.keylevel - 1) * hlp$C_indent
1506      2222 5                      ELSE IF ((.flags AND hlp$M_nohtxt) NEQ 0)
1507      2223 5                          AND (.keylevel EQL 0)
1508      2224 4                          THEN spaces = hlp$C_indent
1509      2225 4                          ELSE spaces = .keylevel * hlp$C_indent;
1510      2226 4
1511      2227 4          ptr = CH$FILL (%ASCII ' ', .spaces, output_buf);
1512      2228 5          IF ((.true_keys EQL 0) AND NOT .help_flags [hlp$V_help])
1513      2229 5              THEN BEGIN
1514      2230 5                  true_keys = -1;
1515      2231 5                  output_desc [dsc$W_length] = .topics_available [0];
1516      2232 5                  CH$MOVE (.output_desc [dsc$W_length], topics_available + 1, .ptr);
1517      2233 5
1518      2234 5          ELSE BEGIN
1519      2235 5              CH$MOVE (.output_desc [dsc$W_length], .linedesc [dsc$A_pointer], .ptr);
1520      2236 4              END;
1521      2237 4          output_desc [dsc$W_length] = .spaces + .output_desc [dsc$W_length];
1522      2238 3
1523      2239 3
1524      2240 3
1525      2241 3
1526      2242 2
1527      2243 2
1528      2244 2      RETURN .status;
1529      2245 1      END;                                !Of output_driver

```

.PSECT S0N\$NOEXE,2

16 00054 TOPICS_AVAILABLE:

:BYTE 22

:ASCII \Information available:\

61 76 61 20 6E 6F 69 74 61 6D 72 6F 66 6E 49 00055

:

3A 65 6C 62 61 6C 69 00064

.PSECT SCODE\$,NOWRT,2							
OFFC 00000 OUTPUT_DRIVER:							
							.WORD
	5E	FDF8	CE	9E 00002	MOVAB	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 2084
	57	0C	AC	D0 00007	MOVL	-520(SP), SP	: 2122
	59	12	A7	9E 0000B	MOVAB	PRINTDATA, R7	: 2124
			01	D0 0000F	PUSHL	18(R7), R9	: 2142
0A	5A	08	BC	D0 00011	MOVL	#1	: 2143
	69		01	E0 00015	BBS	@HELPFLAGS, FLAGS	: 2145
	07		5A	E9 00019	BLBC	FLAGS, 1\$: 2146
	69		04	8A 0001C	BICB2	#4, (R9)	: 2148
	50		01	D0 0001F	MOVL	#1, R0	: 2149
			04	00022	RET		
	69		06	88 00023	1\$: BISB2	#6, (R9)	: 2153
	56	10	BC	D0 00026	MOVL	@HELPLEVEL, KEYLEVEL	: 2155
		04	AE	D4 0002A	CLRL	OUTPUT DESC	: 2156
	04	58	04	AC D0 0002D	MOVL	LINEDESC, R8	: 2157
	08	AE	0C	AE 9E 00031	MOVW	(R8), OUTPUT DESC	: 2158
			5A	D5 0003A	MOVAB	OUTPUT_BUF, OUTPUT_DESC+4	: 2159
			05	12 0003C	TSTL	FLAGS	: 2160
	0000'	CF		08 88 0003E	BNEQ	2\$: 2161
	48		5A	5B D4 00043	2\$: BISB2	#8 PROMPT_FLAGS	: 2168
			01	E1 00045	CLRL	R1\$	
	40	0000'	CF	5B D6 00049	BBC	#1 FLAGS, 5\$: 2169
	07	0000'	CF	01 E0 0004B	INCL	R1\$: 2171
	0000'	CF	03	E1 00051	BBS	#1, PROMPT_FLAGS, 5\$: 2172
			02	88 00057	BBBC	#3, PROMPT_FLAGS, 3\$	
			2F	11 0005C	BISB2	#2, PROMPT_FLAGS	
56	11	A7	08	00 ED 0005E	BRB	4\$	
			00	13 00064	CMPZV	#0, #8, 17(R7), KEYLEVEL	: 2174
			27	13 00066	BEQL	4\$	
	00	B7		22 13 00069	TSTW	OUTPUT_DESC	: 2176
	11	A7	04	56 90 0006B	BEQL	4\$	
	04	B8	04	AE 28 0006F	MOVAB	KEYLEVEL, 17(R7)	: 2178
	50		11	A7 9A 00076	MOVC3	OUTPUT_DESC, @4(R8), @0(R7)	: 2179
	0C B740		04	AE 3C 0007A	MOVZBL	17(R7), R0	
	50		04	AE 3C 00080	MOVZWL	OUTPUT_DESC, @12(R7)[R0]	: 2180
	67		50	C0 00084	ADDL2	OUTPUT_DESC, R0	
	00	B7	20	90 00087	MOVB	R0 (R7)	
			67	D6 0008B	INCL	#3\$, @0(R7)	: 2181
	04	A7	56	D0 0008D	4\$: MOVL	(R7\$)	: 2182
			51	D4 00091	CLRL	KEYLEVEL, 4(R7)	: 2185
			0D	E9 00093	R1	R1	: 2193
	0000'	CF	51	D6 00096	BLBC	FLAGS, 6\$	
			02	88 00098	INCL	R1	
	03		69	E8 0009D	BISB2	#2, PROMPT_FLAGS	: 2195
	69		01	88 000A0	BLBS	(R9), 6\$: 2196
07			50	D4 000A3	BISB2	#1, (R9)	: 2197
	5A		02	E1 000A5	CLRL	RO	: 2204
	0000'	CF	50	D6 000A9	BBC	#2, FLAGS, 7\$	
			04	88 000AB	INCL	RO	
					BISB2	#4, PROMPT_FLAGS	: 2205

LBR_OUTPUTHELP V04=000	Prompting and library searching help function Routine output_driver		H 16	16-Sep-1984 02:04:00	14-Sep-1984 12:37:45	VAX-11 Bliss-32 V4.0-742 [LBR.SRC]OUTPUTHLP.B32;1	Page 50 (13)
		10	A7 95 000B0	7\$: TSTB 09 12 000B3 BNEQ 05 E0 000B5 BBS 50 E9 000B8 BLBC 56 D5 000BE 8\$: TSTL 04 14 000C0 BGTR 01 E0 000C2 BBS 68 B5 000C6 9\$: TSTW 4C 13 000C8 BEQL 5B E9 000CA BLBC A6 9E 000CD MOVAB 02 A5 000D1 MULW3 10 11 000D5 BRB 51 E9 000D7 10\$: BLBC 56 D5 000DA TSTL 05 12 000DC BNEQ 02 B0 000DE MOVW 04 11 000E1 BRB 02 A5 000E3 MULW3 00 2C 000E7 12\$: MOVCS AE 000EC TSTB A7 95 000EE 13\$: BNEQ 19 12 000F1 BBS 05 E0 000F3 #5, HELP FLAGS, 13\$ 01 8E 000F9 MNEG.B 04 AE 0000' CF 9B 000FD MOVZBW 63 0000' CF 04 AE 23 00103 MOVCS 04 AE 06 1 0010A BRB 04 B8 04 AE 28 0010C 13\$: MOVCS 04 AE 5A A0 00112 14\$: ADDW2 08 B7 04 AE 9F 00116 15\$: PUSHAB 6E 50 D0 0011D CALLS 50 6E D0 00120 16\$: MOVL 04 00123 RET	16(R7) 8\$ #5, HELP_FLAGS, 8\$ R0, 16\$ KEYLEVEL #1 FLAGS, 16\$ (R8) 15\$ R11, 10\$ -1(R6), R0 #2 R0, SPACES 12\$ R1, 11\$ KEYLEVEL 11\$ #2 SPACES 12\$ #2, KEYLEVEL, SPACES #0, (SP), #32, SPACES, OUTPUT_BUF	: 2211 : 2212 : 2213 : 2214 : 2215 : 2217 : 2220 : 2221 : 2222 : 2223 : 2224 : 2225 : 2227 : 2228 : 2230 : 2231 : 2232 : 2234 : 2235 : 2237 : 2240 : 2244 : 2245	
03	0000' CF 62						
5A	5A						
5A	0A 50 50	FF					
5A	09						
5A	5A						
5A	5A 20	56 6E	OC				
13	0000' CF 10 A7 04 AE	0000' CF 04	10				
63	0000' CF						
63	04 B8 04 AE 08 B7						

: Routine Size: 292 bytes, Routine Base: \$CODE\$ + 0ABA

```
1531 2246 1 %SBTTL 'Routine libs_available';
1532 2247 1 ROUTINE libs_available (outputroutine, outputwidth) =
1533 2248 2 BEGIN
1534 2249 2
1535 2250 2
1536 2251 2 ++ FUNCTIONAL DESCRIPTION
1537 2252 2
1538 2253 2 Output a list of the define default libraries.
1539 2254 2
1540 2255 2 INPUTS:
1541 2256 2
1542 2257 2     outputroutine = address of the user supplied output routine
1543 2258 2
1544 2259 2     outputwidth = address of a longword containing the output line width
1545 2260 2
1546 2261 2 OUTPUTS:
1547 2262 2
1548 2263 2     The list is output by using the user supplied routine.
1549 2264 2
1550 2265 2 ROUTINE VALUE:
1551 2266 2
1552 2267 2     Always true.
1553 2268 2
1554 2269 2
1555 2270 2 LOCAL
1556 2271 2     acmode,
1557 2272 2     blank_line : BBLOCK [dsc$C_s_bln],
1558 2273 2     filespec : BBLOCK [dsc$C_s_bln],
1559 2274 2     filespec_buf : BBLOCK [nam$C_maxrss],
1560 2275 2     filename : BBLOCK [dsc$C_s_bln],
1561 2276 2     filename_buf : VECTOR [filename_length, BYTE],
1562 2277 2     libno,
1563 2278 2     output_desc : BBLOCK [dsc$C_s_bln],
1564 2279 2     output_buf : BBLOCK [hlp$C_pagesize],
1565 2280 2     status;
1566 2281 2
1567 2282 2 OWN
1568 2283 2     header : COUNTEDSTRING (' Additional help libraries available (type @name for topics):');
1569 2284 2
1570 2285 2     filespec = 0;                                ! Init file spec desc
1571 2286 2     filespec [dsc$A_pointer] = filespec_buf;
1572 2287 2
1573 2288 2     libno = -1;                                ! Init search
1574 2289 2     status = false;                            ! Init status
1575 2290 2
1576 2291 2 WHILE NOT .status                         ! Loop searching for a library
1577 2292 2 DO BEGIN
1578 2293 2     IF NOT tran_next_lib (filespec, acmode, libno) ! Get first lib spec
1579 2294 2         THEN RETURN true;                  ! If none, then return
1580 2295 2     status = file_present (filespec);        ! Is file present?
1581 2296 2 END;
1582 2297 2
1583 2298 2     filename = 0;                            ! Init file name desc
1584 2299 2     filename [dsc$A_pointer] = filename_buf;
1585 2300 2
1586 2301 2     blank_line = 0;                          ! Init blank line desc
1587 2302 2     blank_line [dsc$A_pointer] = output_buf;
```

```

1588 2303 2
1589 2304 2 output_desc = .header [0];
1590 2305 2 output_desc [dsc$w_pointer] = header [1];           ! Init header desc
1591 2306 2 (.outputroutine) (blank_line);
1592 2307 2 (.outputroutine) (output_desc);                      ! Output header
1593 2308 2 (.outputroutine) (blank_line);
1594 2309 2 (.outputroutine) (blank_line);
1595 2310 2
1596 2311 2 output_desc = 0;                                     ! Reuse output desc for
1597 2312 2 output_desc [dsc$w_pointer] = output_buf;          ! lib name lists
1598 2313 2 CHSFILE ('XX'20', ..outputwidth, output_buf);
1599 2314 2
1600 2315 2 switch_libname (filespec, filename);                ! Get first lib name
1601 2316 2 output_desc [dsc$w_length] = .filename[dsc$w_length] + 2; ! Move it into
1602 2317 2 CHSMOVE (.filename[dsc$w_length], filename_buf, output_buf + 2); ! the list.
1603 2318 2
1604 2319 2 WHILE tran_next_lib(filespec, acmode, libno) DO      ! While more libs translate,
1605 2320 2   IF file_present(filespec) THEN                      ! if the file exists,
1606 2321 3     BEGIN
1607 2322 3       switch_libname(filespec, filename);             ! get its name.
1608 2323 3
1609 2324 3   ! If we can't fit a couple of spaces and the filename on the
1610 2325 3   ! line we're working on, send the line to the output routine and
1611 2326 3   ! start a new one:
1612 2327 3
1613 2328 3   IF (.output_desc[dsc$w_length] + 2 + .filename[dsc$w_length]) GTRU ..outputwidth THEN
1614 2329 4     BEGIN
1615 2330 4       (.outputroutine) (output_desc);
1616 2331 4       output_desc [dsc$w_length] = 0;
1617 2332 4       CHSFILE ('XX'20', ..outputwidth, output_buf);
1618 2333 3     END;
1619 2334 3
1620 2335 3   ! Move the filename onto the line (leaving two padding spaces):
1621 2336 3
1622 2337 3   CHSMOVE (.filename[dsc$w_length], filename_buf,
1623 2338 3           output_buf + .output_desc[dsc$w_length] + 2);
1624 2339 3   output_desc [dsc$w_length] = .output_desc[dsc$w_length] + 2 + .filename[dsc$w_length];
1625 2340 2
1626 2341 2   (.outputroutine) (output_desc);                   ! Output last line built
1627 2342 2
1628 2343 2   RETURN true;
1629 2344 1 END;

```

.PSECT \$0WNS,NOEXE,2

65 68 20 6C 61 6E 6F 69 74 69 64 64 41 20 3E 0006B	0006C HEADER: .BLKB 1
76 61 20 73 65 69 72 61 72 62 69 6C 20 70 6C 0007C	.BYTE 62
74 20 72 6F 66 20 65 6D 61 6E 40 20 65 70 79 0008B	.ASCII '\ Additional help libraries available (t\
3A 29 73 63 69 70 6F 000A4	.ASCII '\ type a name for topics):\

.PSECT SCODES,NOWRT,2

03FC 00000 LIBS_AVAILABLE:

LBR_OUTPUTHELP
V04=000

Prompting and library searching help function
Routine libs_available

L 16
16-Sep-1984 02:04:00
14-Sep-1984 12:37:45

VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1

Page 54
(14)

		08 BC	50 D1 000D4	CMPL	R0, \$OUTPUTWIDTH	
		68 FEB8	13 1B 000D8	BLEQU	\$	
		68 FEB8	CD 9F 000DA	PUSHAB	OUTPUT DESC	2330
		6E FEB8	01 FB 000DE	CALLS	#1 (R8)	2331
	08 BC	6E FEB8	CD B4 000E1	CLRW	OUTPUT DESC	2332
	20	08 AE	00 2C 000E5	MOVCS	#0, (SP), #32, \$OUTPUTWIDTH, OUTPUT_BUF	
	0A AE46	FEC0 CD	56 FEB8 CD 3C 000ED	MOVZWL	OUTPUT DESC, R6	2338
		50 FEB8 CD	57 28 000F2	MOVCS3	R7, F1[ENAMÉ BUF, OUTPUT_BUF+2[R6]]	
		50 FEB8 CD	02 A746 9E 000FA	MOVAB	2(R7)[R6], R0	2339
		50 FEB8 CD	50 B0 000FF	MOVW	R0, OUTPUT_DESC	
		50 FEB8 CD	9B 11 00104	BRB	4\$	2320
		68 FEB8	CD 9F 00106	PUSHAB	OUTPUT DESC	2342
		68 FEB8	01 FA 0010A	CALLS	#1, (R8)	
		50 FEB8 CD	01 D0 0010D	MOVL	#1, R0	2343
		50 FEB8 CD	04 00110	RET		2344

: Routine Size: 273 bytes. Routine Base: \$CODE\$ + 0BDE

```

1631      2345 1 XSBTTL 'Routine file_present';
1632      2346 1 ROUTINE file_present(filename) =
1633      2347 2 BEGIN
1634      2348 2
1635      2349 2 ++ FUNCTIONAL DESCRIPTION:
1636      2350 2
1637      2351 2
1638      2352 2 Return success if the file exists.
1639      2353 2
1640      2354 2 INPUTS:
1641      2355 2
1642      2356 2 filename = address of desc of file name
1643      2357 2
1644      2358 2 OUTPUTS:
1645      2359 2
1646      2360 2
1647      2361 2
1648      2362 2
1649      2363 2 ROUTINE VALUE:
1650      2364 2
1651      2365 2 True, if file exists
1652      2366 2 False, if it doesn't
1653      2367 2
1654      2368 2
1655      2369 2 MAP
1656      2370 2 filename : REF BBLOCK;
1657      2371 2
1658      2372 2 LOCAL
1659      2373 2
1660      2374 2
1661      2375 2
1662      2376 2
1663      2377 2
1664      P 2378 2 SNAME_INIT ( NAM = nam,
1665      P 2379 2
1666      P 2380 2 ESS = nam$c_maxrss,
1667      P 2381 2
1668      P 2382 2 ESA = string);
1669      P 2383 2
1670      P 2384 2 SFAB_INIT ( FAB = fab,
1671      P 2385 2
1672      P 2386 2
1673      P 2387 2
1674      P 2388 2
1675      P 2389 2 IF (status = $PARSE (FAB = fab))
1676      P 2390 2
1677      P 2391 2 THEN (status = $SEARCH (FAB = fab));
1678      P 2392 2
1679      P 2393 1 RETURN .status;
1 END;

```

.EXTRN SYSSSEARCH

003C 00000 FILE_PRESENT:	
5E FE50 CE 9E 00002	.WORD MOVAB
	Save R2,R3,R4,R5
	-432(SP), SP

: 2346
:

LBR.OUTPUTHELP B 1
 V04=000 Prompting and library searching help function 16-Sep-1984 02:04:00
 Routine file_present 14-Sep-1984 12:37:45 VAX-11 Bliss-32 V4.0-742
 [LBR.SRC]OUTPUTHELP.B32:1 Page 56
 (15)

0060	BF	.	00	6E	FF50	00	2C	00007	MOVCS	#0, (SP), #0, #96, SRMS_PTR	2380
				CD	6002	CD	C000E				
				FF5A	CD	8F	B0	00011	MOVW	#24578, SRMS_PTR	
				FF5C	CD	01	8E	00018	MNEGB	#1, SRMS_PTR+10	
0050	BF	00		6E	AD	6E	9E	0001D	MOVAB	STRING, SRMS_PTR+12	2387
					AD	00	2C	00022	MOVCS	#0, (SP), #0, #80, SRMS_PTR	
					5003	AD	00029				
				B0	AD	8F	B0	0002B	MOVW	#20483, SRMS_PTR	
				C6	AD	02	90	00031	MOVB	#2, SRMS_PTR+22	
				CF	AD	02	90	00035	MOVB	#2, SRMS_PTR+31	
				D8	AD	FF50	CD	00039	MOVAB	NAM, SRMS_PTR+40	
				50	04	AC	D0	0003F	MOVL	FILÉNAME, R0	
				DC	AD	04	A0	00043	MOVL	4(R0), SRMS_PTR+44	2389
				E0	AD	0000	CF	00048	MOVAB	SYSHELP+1, SRMS_PTR+48	
				E4	AD	0000	60	0004E	MOVB	(R0), SRMS_PTR+52	
				E5	AD	0000	CF	00052	MOVB	SYSHELP, SRMS_PTR+53	
					B0	AD	9F	00058	PUSHAB	FAB	
				00000000G	00	01	FB	0005B	CALLS	#1, SY\$PARSE	
					0A	50	E9	00062	BLBC	STÁTUS, 1\$	
				00000000G	00	B0	AD	9F	PUSHAB	FAB	2390
						01	FB	00065	CALLS	#1, SY\$SEARCH	
						04	0006F	1\$:	RET		2393

: Routine Size: 112 bytes. Routine Base: \$CODE\$ + 0CEF

```

: 1681 2394 1 %SBTTL 'Routine nohelp_log';
: 1682 2395 1 ROUTINE nohelp_log (logdesc) =
: 1683 2396 2 BEGIN
: 1684 2397 2  ++
: 1685 2398 2  | FUNCTIONAL DESCRIPTION:
: 1686 2399 2  |
: 1687 2400 2  | If the logical name for a log file is defined, then put a record
: 1688 2401 2  | into the specified log file. If that file does not already exist,
: 1689 2402 2  | then create it.
: 1690 2403 2  |
: 1691 2404 2  |
: 1692 2405 2  | INPUTS:
: 1693 2406 2  |
: 1694 2407 2  |     logdesc = address of string descriptor for record to be output
: 1695 2408 2  |
: 1696 2409 2  | OUTPUTS:
: 1697 2410 2  |
: 1698 2411 2  |     None.
: 1699 2412 2  |
: 1700 2413 2  | ROUTINE VALUE:
: 1701 2414 2  |
: 1702 2415 2  |     Always true.
: 1703 2416 2  |
: 1704 2417 2  |
: 1705 2418 2  |
: 1706 2419 2  |
: 1707 2420 2  | MAP
: 1708 2421 2  |     logdesc : REF BBLOCK;
: 1709 2422 2  | LOCAL
: 1710 2423 2  |     logresult : VECTOR [nam$c_maxrss, BYTE],
: 1711 2424 2  |     logrsdesc : BBLOCK [dsc$c_s_bln],
: 1712 2425 2  |     lognam : BBLOCK [nam$c_bln],
: 1713 2426 2  |     logfab : BBLOCK [fab$c_bln],
: 1714 2427 2  |     lograb : BBLOCK [rab$c_bln],
: 1715 2428 2  |     logfile : BBLOCK [dsc$c_s_bln],
: 1716 2429 2  |     logfiletrn : BBLOCK [dsc$c_s_bln],
: 1717 2430 2  |     status;
: 1718 2431 2  |
: 1719 2432 2  | OWN
: 1720 2433 2  |     logstring : COUNTEDSTRING ('HELP$LOG');
: 1721 2434 2  |
: 1722 2435 2  |     logfile [dsc$w_length] = .logstring [0];
: 1723 2436 2  |     logfile [dsc$sa_pointer] = logstring [1];
: 1724 2437 2  |
: 1725 2438 2  |     logfiletrn [dsc$w_length] = nam$c_maxrss;
: 1726 2439 2  |     logfiletrn [dsc$sa_pointer] = logresult;
: 1727 2440 2  |
: 1728 P 2441 2  | $NAM_INIT ( NAM = lognam,
: 1729 P 2442 2  |           ESS = nam$c_maxrss,
: 1730 P 2443 2  |           ESA = logresult,
: 1731 P 2444 2  |           RSS = nam$c_maxrss,
: 1732 P 2445 2  |           RSA = logresult);
: 1733 2446 2  |
: 1734 P 2447 2  | $FAB_INIT ( FAB = logfab
: 1735 P 2448 2  |           FNS = .logfile [dsc$w_length],
: 1736 P 2449 2  |           FNA = .logfile [dsc$sa_pointer],
: 1737 P 2450 2  |           FAC = PUT,
```

!Descriptor of log record

!Space for HELP\$LOG resultant filename

!Descriptor for result name

!NAM block for HELP\$LOG

!FAB for output to HELP\$LOG

!RAB for output to HELP\$LOG

!Descriptor for HELP\$LOG translation

! Initialize logical name

! Initialize descriptor for logical name translation

! Initialize name block

! Initialize fab

LBR_OUTPUTHELP Prompting and library searching help function
V04-000 Routine nohelp_log

D 1
16-Sep-1984 02:04:00
14-Sep-1984 12:37:45
VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1

Page 58
(16)

```

1738 P 2451 2      FOP = CIF.
1739 P 2452 2      RAT = CR,
1740     25      NAM = Lognam);
1741 P 2454 2      $RAB_INIT ( RAB = Lograb,
1742     25      FAB = Logfab,
1743     25      ROP = EOF);           ! Initialize rab
1744
1745 P 2458 2
1746 P 2459 2
1747 P 2460 2      ! If HELPSLOG can be successfully translated,
1748     25      1. Create the file if it doesn't already exist.
1749     25      2. Connect to that file.
1750     25      3. Write the record to that file.
1751     25      4. Clean up afterwards.
1752 P 2465 2
1753 P 2466 2
1754 P 2467 4      IF ((status = $STRNLOG (LOGNAM = Logfile, RSLBUF = logfiletrn))
1755     3      AND (.status NEQ SSS_NOTRAN)) THEN IF (status = $CREATE (FAB = logfab))
1756     3      THEN BEGIN
1757     4      IF (status = $CONNECT (RAB = Lograb))
1758     4      THEN BEGIN
1759     4          Lograb [rab$w_rsz] = .logdesc [dsc$w_length];
1760     4          Lograb [rab$1_rbf] = .logdesc [dsc$1a_pointer];
1761     4          logrsdesc [dsc$w_length] = .lognam [nam$b_rsl];
1762     4          logrsdesc [dsc$1a_pointer] = .lognam [nam$1_rsa];
1763     4          SPUT (RAB = Lograb);
1764     5          IF NOT (status = $DISCONNECT (RAB = Lograb))
1765     4          THEN SIGNAL ((shr$closeout OR hlp$c_facility OR sts$k_warning),
1766     4                  1, logrsdesc, .status, .lograb [rab$1_stv]);
1767     3      END;
1768     4      IF NOT (status = $CLOSE (FAB = logfab))
1769     3      THEN SIGNAL ((shr$closeout OR hlp$c_facility OR sts$k_warning),
1770     3                  1, logrsdesc, .status, .lograb [rab$1_stv]);
1771     2      END;
1772 P 2485 2      RETURN true
1773 P 2486 1      END;           !Of nohelp_log

```

```

.PSECT $OWNS,NOEXE,2
        000AB .BLKB   1
        08 000AC LOGSTRING:
        47 4F 4C 24 50 4C 45 48 000AD  :BYTE   8
                                         :ASCII   \HELP$LOG\

                                         .EXTRN  SY$CREATE, SY$CONNECT
                                         .EXTRN  SY$PUT, SY$DISCONNECT
                                         .EXTRN  SY$CLOSE

.PSECT $CODE$,NOWRT,2
        007C 00000 NOHELP_LOG:
        08 56 0000000G 00 9E 00002  .WORD   Save R2,R3,R4,R5,R6
                                         MOVAB  LIB$SIGNAL, R6
                                         SE FDF4 CE 9E 00009  MOVAB  -524(SP), SP
                                         0000' CF 9B 0000E  MOVZBW LOGSTRING, LOGFILE

```

: 2395

: 2435

LBR_OUTPUTHELP		Prompting and library searching help function			E 1 16-Sep-1984 02:04:00		VAX-11 Bliss-32 V4.0-742 [LBR.SRC]OUTPUTHELP.B32;1	Page 59 (16)
VO4=000	Routine nohelp_log				14-Sep-1984 12:37:45			
0060	BF	00	0C AE 0000' FF 8F 9E 00014 04 AE FF00 CD 9E 0001A 6E 00 2C 00024 00A4 CE 00A4 8F B0 0002E 00A6 CE 6002 01 8E 00035 00AB CE FF00 CD 9E 0003A 00AE CE 00B0 FF00 CD 9E 00041 0080 CE 0080 00 2C 0004D 54 AE 54 AE 5003 8F B0 00056 58 AE 02000000 8F D0 0005C 6A AE 0202 01 90 00064 72 AE 00A4 8F B0 00068 7C AE 0080 CE 9E 0006E 0088 CE 0088 OC AE 00074 08 AE 08 AE 90 0007A 6E 00 2C 00080 10 AE 10 AE 4401 8F B0 00089 14 AE 0100 8F 3C 0008F 4C AE 54 AE 9E 00095 7E 7C 0009A 7E D4 0009C 0C AE 9F 0009E 1C AE 9F 000A1 00000000G 00 06 FB 000A6 52 50 D0 000AD 09 52 E9 000B0 00000629 8F 52 D1 000B3 03 12 000BA 54 008D 31 000BC 1\$: 00000000G 00 54 AE 9F 000BF 2\$: 52 01 FB 000C2 7D 50 D0 000C9 10 AE 9F 000CC 00000000G 00 52 E9 000CF 52 01 FB 000D2 49 50 D0 000D9 32 AE 04 AC D0 000DF 38 AE 04 AO D0 000E7 0104 CE 00A7 CE 9B 000EC FEFC CD 00A8 CE D0 000F3 10 AE 9F 000FA 00000000G 00 10 01 FB 000FD 00000000G 00 10 AE 9F 00104 52 01 FB 00107 14 50 D0 0010E 1C AE 98 00111 FEF8 CD 9F 00119 FF 0002B MOVAB LOGSTRING+1, LOGFILE+4 MOVZBW #255, LOGFILETRN MOVAB LOGRESULT, LOGFILETRN+4 MOVCS #0, (SP), #0, #96, SRMS_PTR MOVW #24578, SRMS_PTR MNEGCB #1, SRMS_PTR+2 MOVAB LOGRESULT, SRMS_PTR+4 MNEGCB #1, SRMS_PTR+10 MOVAB LOGRESULT, SRMS_PTR+12 MOVCS #0, (SP), #0, #80, SRMS_PTR MOVW #20483, SRMS_PTR MOVL #33554432, SRMS_PTR+4 MOVAB #1, SRMS_PTR+22 MOVW #514, SRMS_PTR+30 MOVAB LOGNAM, SRMS_PTR+40 MOVL LOGFILE+4, SRMS_PTR+44 MOVAB LOGFILE, SRMS_PTR+52 MOVCS #0, (SP), #0, #68, SRMS_PTR MOVW #17409, SRMS_PTR MOVZWL #256, SRMS_PTR+4 LOGFAB, SRMS_PTR+60 CLRQ -(SP) CLRL -(SP) PUSHAB LOGFILETRN CLRL -(SP) PUSHAB LOGFILE CALLS #6, SYS\$TRNLOG MOVL R0, STATUS BLBC STATUS, 1\$ CMPL STATUS, #1577 BNEQ 2\$ BRW 4\$ PUSHAB LOGFAB CALLS #1, SYSSCREATE MOVL R0, STATUS BLBC STATUS, 4\$ PUSHAB LOGRAB CALLS #1, SYSSCONNECT MOVL R0, STATUS BLBC STATUS, 3\$ MOVL LOGDESC, R0 (R0), LOGRAB+34 4(R0), LOGRAB+40 MOVZBW LOGNAM+3, LOGRSDESC MOVL LOGNAM+4, LOGRSDESC+4 PUSHAB LOGRAB CALLS #1, SYS\$PUT PUSHAB LOGRAB CALLS #1, SYS\$DISCONNECT MOVL R0, STATUS BLBS STATUS, 3\$ PUSHL LOGRAB+12 PUSHL STATUS PUSHAB LOGRSDESC	: 2436 : 2438 : 2439 : 2445 : 2453 : 2457 : 2467 : 2468 : 2470 : 2472 : 2473 : 2474 : 2475 : 2476 : 2477 : 2478 : 2479				

LBR.OUTPUTHELP
V04-000

Prompting and library searching help function
Routine nohelp_log

F 1
16-Sep-1984 02:04:00
14-Sep-1984 12:37:45

VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32:1

Page 60
(16)

		01	DD	0011D	PUSHL	#1	
		8F	DD	0011F	PUSHL	#7737432	
66		05	FB	00125	CALLS	#5, LIB\$SIGNAL	
	54	AE	9F	0012B	PUSHAB	LOGFAB	2481
00000000G	00	01	FB	0012B	CALLS	#1, SYSSCLOSE	
52		50	DO	00132	MOVL	R0, STATUS	
14		52	E8	00135	BLBS	STATUS, 4\$	
	1C	AE	DD	00138	PUSHL	LOGRAB+12	2483
		52	DD	0013B	PUSHL	STATUS	
	FEF8	CD	9F	0013D	PUSHAB	LOGRSDESC	2482
		01	DD	00141	PUSHL	#1	
		8F	DD	00143	PUSHL	#7737432	
66		05	FB	00149	CALLS	#5, LIB\$SIGNAL	
50		01	DO	0014C	PUSHL	#1, R0	2485
		04	0014F	4\$:	RET		2486

; Routine Size: 336 bytes. Routine Base: \$CODE\$ + 0D5F

00FC 00000 REMOVE_LAST_KEY:

57	04	AC	D0	00002	.WORD	Save R2,R3,R4,R5,R6,R7
56		67	3C	00006	MOVL	STRINGDESCR, R7
56	04	A7	C0	00009	MOVZWL	(R7), R6
56	08	AC	C2	0000D	ADDL2	4(R7), R6
56		DC	C2	00011	SUBL2	LAST_KEY LENGTH, R6
56	50	00000'	CF	9A	SUBL2	#12 LAST_CHAR
01	A6	0000'	CF	50	MOVZBL	SUBTOPIC, R0
67		56	04	28	MOVCL3	R0, SUBTOPIC+1, 1(LAST_CHAR)
		56		00019	SUBL2	4(R7), R6
		56	0B	A1	ADDW3	#11, R6, (R7)
		56		00024		

LBR.OUTPUTHELP Prompting and library searching help function
V04=000 Routine remove_last_key

H 1
16-Sep-1984 02:04:00
14-Sep-1984 12:37:45

VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32:1

Page 62
(17)

50 01 D0 00028
04 0002B MOVL #1, R0
RET

; 2526
; 2527

: Routine Size: 44 bytes, Routine Base: \$CODE\$ + 0EAF

LBR
V04

```

1817 2528 1 %SBTTL 'Routine remove_terminator';
1818 2529 1 ROUTINE remove_terminator (stringdescr) =
1819 2530 2 BEGIN
1820 2531 2
1821 2532 2+++
1822 2533 2||| FUNCTIONAL DESCRIPTION:
1823 2534 2||| Remove the termination characters at the end of the string descriptor.
1824 2535 2|||
1825 2536 2||| INPUTS:
1826 2537 2|||     stringdescr = address of string descriptor for input text string
1827 2538 2|||
1828 2539 2||| OUTPUTS:
1829 2540 2|||     stringdescr = input descriptor with termination characters removed
1830 2541 2|||
1831 2542 2||| ROUTINE VALUE:
1832 2543 2|||     Always true.
1833 2544 2|||
1834 2545 2|||
1835 2546 2|||
1836 2547 2|||
1837 2548 2|||
1838 2549 2|||
1839 2550 2|||
1840 2551 2||| MAP
1841 2552 2|||     stringdescr : REF BBLOCK;
1842 2553 2|||
1843 2554 3 WHILE (CH$RCHAR (.stringdescr [dsc$w_pointer]
1844 2555 3           + .stringdescr [dsc$w_length] - 1) LSS %X'20') ! While termination character present
1845 2556 3           AND (.stringdescr [dsc$w_length] GTR 0)
1846 2557 2 DO stringdescr [dsc$w_length] := .stringdescr [dsc$w_length] - 1;! Remove terminator
1847 2558 2
1848 2559 2 RETURN true;
1849 2560 1 END;                                ! Of remove_terminator

```

0000 00000 REMOVE_TERMINATOR:

				.WORD	Save nothing	: 2529
51	04	AC	D0	00002	MOVL STRINGDESCR, R1	: 2554
50	04	BC	3C	00006 1\$:	MOVZWL @STRINGDESCR, R0	: 2555
50	04	A1	C0	0000A	ADDL2 4(R1), R0	
20	FF	A0	91	0000E	CMPB -1(R0), #32	
		0A	1E	00012	BGEQU 2\$	
	04	BC	B5	00014	TSTW @STRINGDESCR	: 2556
	05	13	00017		BEQL 2\$	
	04	BC	B7	00019	DECW @STRINGDESCR	: 2557
		E8	11	0001C	BRB 1\$	
50		01	D0	0001E 2\$:	MOVL #1, R0	: 2559
		04	00021		RET	: 2560

: Routine Size: 34 bytes, Routine Base: SCODES + 0EDB

```

1851 2561 1 %SBTTL 'Routine make_upper_case'.
1852 2562 1 ROUTINE make_upper_case (idesc: REF BLOCK[, BYTE], oname: REF VECTOR[, BYTE]) =
1853 2563 2 BEGIN
1854 2564 2
1855 2565 2 /**
1856 2566 2 FUNCTIONAL DESCRIPTION:
1857 2567 2
1858 2568 2 Upper case the name described by string descriptor idesc and
1859 2569 2 put the name at location oname. (Also substitutes a space
1860 2570 2 character for a horizontal tab.)
1861 2571 2
1862 2572 2 INPUTS:
1863 2573 2
1864 2574 2 idesc = address of string descriptor for input text string
1865 2575 2
1866 2576 2 oname = address of buffer to contain uppercase output string
1867 2577 2
1868 2578 2 OUTPUTS:
1869 2579 2
1870 2580 2 oname : as described above
1871 2581 2
1872 2582 2 ROUTINE VALUE:
1873 2583 2
1874 2584 2 Always true.
1875 2585 2
1876 2586 2 /**
1877 2587 2
1878 2588 2 LITERAL
1879 2589 2 fill= 0;
1880 2590 2
1881 2591 2 OWN
1882 2592 2     uppercase_table: VECTOR [256, BYTE] INITIAL (BYTE (
1883 2593 2     0, 1, 2, 3, 4, 5, 6, 7, 8, 32, 10, 11, 12, 13, 14, 15, ! HT -> space.
1884 2594 2     16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
1885 2595 2     32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
1886 2596 2     48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
1887 2597 2     64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
1888 2598 2     80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95,
1889 2599 2     96, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, ! a-o -> A-O.
1890 2600 2     80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 123, 124, 125, 126, 127, ! p-z -> P-Z.
1891 2601 2     128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143,
1892 2602 2     144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159,
1893 2603 2     160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175,
1894 2604 2     176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191,
1895 2605 2     192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
1896 2606 2     208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223,
1897 2607 2     192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, ! Upcase foreign
1898 2608 2     240, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 254, 255); ! too.
1899 2609 2
1900 2610 2     CH$TRANSLATE (uppercase_table,
1901 2611 2         .idesc[dsc$w_length], .idesc[dsc$w_pointer], fill,
1902 2612 2         .idesc[dsc$w_length], .oname);
1903 2613 2
1904 2614 2 RETURN true
1905 2615 1 END;                                !Of make_uppercase

```

OE	OD	OC	OB	OA	20	08	07	06	05	04	03	02	01	00	000B5	000B8	UPCASE_TABLE:
1D	1C	1B	1A	19	18	17	16	15	14	13	12	11	10	0F	000C7		
2C	2B	2A	29	28	27	26	25	24	23	22	21	20	1F	1E	000D6		
3B	3A	39	38	37	36	35	34	33	32	31	30	2F	2E	2D	000E5		
4A	49	48	47	46	45	44	43	42	41	40	3F	3E	3D	3C	000F4		
59	58	57	56	55	54	53	52	51	50	4F	4E	4D	4C	4B	00103		
68	47	46	45	44	43	42	41	60	5F	5E	5D	5C	5B	5A	00112		
57	56	55	54	53	52	51	50	4F	4E	4D	4C	4B	4A	49	00121		
86	85	84	83	82	81	80	7F	7E	7D	7C	7B	7A	59	58	00130		
95	94	93	92	91	90	8F	8E	8D	8C	8B	8A	89	88	87	0013F		
A4	A3	A2	A1	A0	9F	9E	9D	9C	9B	9A	99	98	97	96	0014E		
B3	B2	B1	B0	AF	AE	AD	AC	AB	AA	A9	A8	A7	A6	A5	0015D		
C2	C1	CO	BF	BE	BD	BC	BB	BA	BB	B7	B6	B5	B4	B3	0016C		
D1	DO	CF	CE	CD	CC	CB	CA	C9	C8	C7	C6	C5	C4	C3	0017B		
CO	DF	DE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	0018A		
CF	CE	CD	CC	CB	CA	C9	C8	C7	C6	C5	C4	C3	C2	C1	00199		
FE	DD	DC	DB	DA	D9	D8	D7	D6	D5	D4	D3	D2	D1	F0	001A8		
														FF	001B7		

.PSECT \$OWNS,NOEXE,2

.BLKB 3
.BYTE

0	1	2	3	4	5	6	7	8	32	10	11	-	
12	13	14	15	16	17	18	19	20	21	-	-	:	
22	23	24	25	26	27	28	29	30	31	-	-	:	
32	33	34	35	36	37	38	39	40	41	-	-	:	
42	43	44	45	46	47	48	49	50	51	-	-	:	
52	53	54	55	56	57	58	59	60	61	-	-	:	
62	63	64	65	66	67	68	69	70	71	-	-	:	
72	73	74	75	76	77	78	79	80	81	-	-	:	
82	83	84	85	86	87	88	89	90	91	-	-	:	
92	93	94	95	96	97	98	99	100	101	-	-	:	
70	71	72	73	74	75	76	77	78	79	-	-	:	
80	81	82	83	84	85	86	87	88	89	-	-	:	
90	123	124	125	126	127	128	129	130	131	-	-	:	
-126	-125	-124	-123	-122	-121	-120	-121	-120	-121	-	-	:	
-119	-118	-117	-116	-115	-114	-113	-112	-111	-110	-	-	:	
-112	-111	-110	-109	-108	-107	-106	-105	-104	-103	-	-	:	
-105	-104	-103	-102	-101	-100	-99	-98	-97	-96	-	-	:	
-97	-96	-95	-94	-93	-92	-91	-90	-89	-88	-	-	:	
-82	-81	-80	-79	-78	-77	-76	-75	-74	-73	-	-	:	
-66	-65	-64	-63	-62	-61	-60	-59	-58	-57	-	-	:	
-50	-49	-48	-47	-46	-45	-44	-43	-42	-41	-	-	:	
-34	-33	-64	-63	-62	-61	-60	-59	-58	-57	-	-	:	
-50	-49	-16	-47	-46	-45	-44	-43	-42	-41	-	-	:	
-2	-1												:

.PSECT \$CODE\$,NOWRT,2

003C 00000 MAKE_UPPER CASE:

0000	CF	00	04	50	04	AC	D0	00002	.WORD	Save R2,R3,R4,R5
						60	2E	00006	MOVL	IDESC, R0
			08	BC		60		0000E	MOVTC	(R0), @4(R0), #0, UPCASE_TABLE, (R0), -
				50		01	D0	00011	MOVL	@ONAME
						04	00014		RET	#1, R0

: 2562
: 2611
: 2612
: 2614
: 2615

; Routine Size: 21 bytes, Routine Base: \$CODE\$ + 0EFD

; 1906 2616 0 END ELUDOM

.EXTRN LIB\$SIGNAL

LBR.OUTPUTHELP Prompting and library searching help function
V04=000 Routine make_upper_case

L 1
16-Sep-1984 02:04:00
14-Sep-1984 12:37:45

VAX-11 Bliss-32 V4.0-742
[LBR.SRC]OUTPUTHLP.B32;1

Page 66
(19)

LBF
VO4

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	440	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODES	3858	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
\$_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	127	1	581	00:01.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:OUTPUTHLP/OBJ=OBJ\$:OUTPUTHLP MSRC\$:OUTPUTHLP/UPDATE=(ENH\$:OUTPUTHLP)

Size: 3858 code + 440 data bytes
Run Time: 01:15.5
Elapsed Time: 02:32.9
Lines/CPU Min: 2079
Lexemes/CPU-Min: 27990
Memory Used: 351 pages
Compilation Complete

0199 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

OLDLIB
LIS

OPENCLOSE
LIS

OUTPUTLP
LIS

BRMSG
LIS

0200 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

TRANSFER
LIS

DATABASE
LIS

PUTCACHE
LIS

LIBRAR

PREFIX
REQ

CROSS
LIS

LIBRARIAN
MAP

SUBS
LIS

FILEIO
LIS

PAIDLBR
LIS

COMPRESS
LIS

EXTRACT
LIS

B
MDL

DELETE
LIS